Licenciatura em Ciências da Computação

COMPUTABILIDADE E COMPLEXIDADE

3. Funções Parciais Recursivas

José Carlos Costa

Dep. Matemática
Universidade do Minho
Braga, Portugal

1º semestre 2025/2026

- Em 1921/22, o matemático alemão David Hilbert (1862-1943) deu um curso sobre fundações da matemática no qual propõe que estas sejam reformuladas de forma rigorosa, partindo da aritmética.
- O programa de Hilbert era que toda a matemática poderia ser reduzida a um número finito de axiomas a partir dos quais qualquer proposição da matemática poderia ser provada.
- Em 1931, o matemático austríaco Kurt Gödel (1906-1978) provou, através do seu Teorema da incompletude, que esta tarefa é impossível.
- Este teorema diz que em qualquer sistema axiomático contendo a aritmética de Peano, existem afirmações verdadeiras que não podem ser demonstradas a partir dos axiomas.

- As *funções recursivas primitivas* são aquelas que podem ser obtidas de certas *funções básicas* usando os operadores de *composição* e *recursão primitiva*.

- A maior parte das funções "usuais" (tais como *adição*, *multiplicação*, *exponenciação*,...) são recursivas primitivas.

- Em "*On the Infinite*" (1925), Hilbert conjeturou que a (agora chamada) função de Ackermann não é recursiva primitiva, mas esse facto só foi provado em 1928 por um seu aluno, o matemático alemão Wilhelm Ackermann (1896-1962), no artigo "*On Hilbert's construction of the real numbers*".

- Todas as funções recursivas primitivas bem como a função de Ackermann são exemplos de funções computáveis.

- Deve-se notar que na demonstração do Teorema da incompletude, Gödel mostrou que existem funções não computáveis.
- Vimos já que a noção de função computável (ou seja, uma função que pode ser calculada por processos algorítmicos) foi formalizada em 1936, por Alan Turing (1912-1954) em "*On Computable Numbers, with an Application to the Entscheidungsproblem*".
- Neste artigo, Turing introduziu o conceito de Máquina de Turing, um modelo abstrato de computação, e estabeleceu as bases para as ciências da computação.
- Como vimos, no artigo, Turing propôs ainda a existência de uma Máquina Universal, que poderia executar qualquer cálculo que uma máquina específica pudesse realizar, e respondeu ao "Entscheidungsproblem" (problema da decisão), demonstrando que não existe um algoritmo geral para decidir se qualquer problema matemático pode ser resolvido.

Definição

As funções *iniciais* ou *básicas* são as seguintes:

1. Funções *zero*: para cada inteiro $k \geq 0$, a função *zero* de aridade k é a função
$$\begin{aligned} zero^k : \quad \mathbb{N}_0^k &\rightarrow \mathbb{N}_0 \\ (x_1, \ldots, x_k) &\mapsto 0 \end{aligned}$$
2. Função *sucessor* : é a função
$$\begin{aligned} s : \mathbb{N}_0 &\rightarrow \mathbb{N}_0 \\ x &\mapsto x + 1 \end{aligned}$$
3. Funções *projeção*: para cada natural k e cada $j \in \{1, \ldots, k\}$, a função de aridade k de *projeção* sobre a componente j é a função
$$\begin{aligned} p_j^k : \quad \mathbb{N}_0^k &\rightarrow \mathbb{N}_0 \\ (x_1, \ldots, x_k) &\mapsto x_j \end{aligned}$$

Definição (operador de composição)

Seja

$$f : \mathbb{N}_0^k \to \mathbb{N}_0$$

uma função k-ária e sejam

$$g_1, \ldots, g_k : \mathbb{N}_0^n \to \mathbb{N}_0$$

k funções n-árias. A função n-ária

$$h : \mathbb{N}_0^n \to \mathbb{N}_0$$

definida, para cada $x_1, \ldots, x_n \in \mathbb{N}_0$, por

$$h(x_1, \ldots, x_n) = f\left(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n)\right)$$

é chamada a *composição* de f com $g_1, \ldots, g_k$, e escreve-se

$$h = f \circ (g_1, \ldots, g_k).$$

Exemplo 1

Seja

$$\begin{aligned} ad : \mathbb{N}_0 \times \mathbb{N}_0 &\to \mathbb{N}_0 \\ (x, y) &\mapsto x + y \end{aligned}$$

a função de adição em $\mathbb{N}_0$ e consideremos as funções

$$\begin{aligned} g_1 : \mathbb{N}_0 &\to \mathbb{N}_0 \\ x &\mapsto x^3 \end{aligned} \qquad \text{e} \qquad \begin{aligned} g_2 : \mathbb{N}_0 &\to \mathbb{N}_0 \\ x &\mapsto 2x \end{aligned} .$$

Então $h = ad \circ (g_1, g_2)$ é a função

$$\begin{aligned} h : \mathbb{N}_0 &\to \mathbb{N}_0 \\ x &\mapsto x^3 + 2x. \end{aligned}$$

Teorema

Se $f : \mathbb{N}_0^k \to \mathbb{N}_0$ e $g_1, \ldots, g_k : \mathbb{N}_0^n \to \mathbb{N}_0$ são funções computáveis, então a função composição $f \circ (g_1, \ldots, g_k)$ é computável.

Problema 3.1

1. Calcule os valores de:
   a) $zero^2(1,5)$;
   b) $p_3^4(0,3,8,2)$;
   c) $p_2^3(5,s(6),22)$;
   d) $s \circ s \circ p_1^1(15)$.
2. Considere as funções $g_1, g_2, g_3 : \mathbb{N}_0^2 \to \mathbb{N}_0$ definidas por

$$g_1(x,y) = x + y, \quad g_2(x,y) = 2xy^2, \quad g_3(x,y) = 5y,$$

   e a função $f : \mathbb{N}_0^3 \to \mathbb{N}_0$ definida por $f(x,y,z) = x + y + z^2$.
   a) Identifique a função $h = f \circ (g_1, g_2, g_3)$.
   b) Calcule $h(1,1)$.

Definição (operador de recursão primitiva)

Sejam $f : \mathbb{N}_0^k \to \mathbb{N}_0$ e $g : \mathbb{N}_0^{k+2} \to \mathbb{N}_0$ duas funções. A função $h : \mathbb{N}_0^{k+1} \to \mathbb{N}_0$ definida, para cada $x_1, \ldots, x_k, y \in \mathbb{N}_0$, por

$$\begin{aligned} h(x_1, \ldots, x_k, 0) &= f(x_1, \ldots, x_k) \\ h(x_1, \ldots, x_k, y+1) &= g\big(x_1, \ldots, x_k, y, h(x_1, \ldots, x_k, y)\big) \end{aligned}$$

é dita a função obtida de f e g por *recursão primitiva* (ou *definida recursivamente* por f e g), e denota-se $h = \mathrm{Rec}(f, g)$.

Teorema

Se $f : \mathbb{N}_0^k \to \mathbb{N}_0$ e $g : \mathbb{N}_0^{k+2} \to \mathbb{N}_0$ são funções computáveis, então a função $h = \mathrm{Rec}(f, g)$ obtida de f e g por recursão primitiva também é computável.

Exemplo 2

Seja h a função obtida por recursão primitiva das funções $f : \mathbb{N}_0 \to \mathbb{N}_0$, $x \mapsto x^2$, e $g : \mathbb{N}_0^3 \to \mathbb{N}_0$, $(x, y, z) \mapsto y + z$. Identifiquemos a função h.

Da definição do operador de recursão primitiva resulta que h é a função, de aridade 2, $h : \mathbb{N}_0^2 \to \mathbb{N}_0$ definida, para cada $x, y \in \mathbb{N}_0$, por

$$\begin{array}{rcl} h(x,0) & = & f(x) \\ h(x,y+1) & = & g\big(x,y,h(x,y)\big). \end{array}$$

Tem-se, assim,

$$\begin{array}{rl} h(x,0) & = x^2 \\ h(x,1) & = g(x,0,h(x,0)) = 0 + h(x,0) = x^2, \\ h(x,2) & = g(x,1,h(x,1)) = 1 + h(x,1) = 1 + x^2, \\ h(x,3) & = g(x,2,h(x,2)) = 2 + h(x,2) = 2 + 1 + x^2, \\ h(x,4) & = g(x,3,h(x,3)) = 3 + h(x,3) = 3 + 2 + 1 + x^2, \\ & \vdots \\ h(x,y) & = x^2 + 1 + 2 + \ldots + (y-1) = x^2 + y(y-1)/2 \quad \text{para } y > 1. \end{array}$$

Como $y(y-1)/2$ é igual a 0 quando $y = 0$ ou $y = 1$, pode escrever-se

$$h(x,y) = x^2 + y(y-1)/2$$

para quaisquer $x, y \in \mathbb{N}_0$.

Exemplo 3

Mostremos que a função de adição

$$\begin{aligned} ad : \mathbb{N}_0 \times \mathbb{N}_0 &\to \mathbb{N}_0 \\ (x, y) &\mapsto x + y \end{aligned}$$

pode ser obtida por recursão primitiva de funções f e g, ou seja, $ad = \text{Rec}(f, g)$. Para tal f e g têm de verificar

$$\begin{aligned} ad(x, 0) &= f(x) \\ ad(x, y + 1) &= g\big(x, y, ad(x, y)\big) \end{aligned}$$

para cada $x, y \in \mathbb{N}_0$. Logo

- $f(x) = ad(x, 0) = x + 0 = x$,
- $g\big(x, y, ad(x, y)\big) = ad(x, y + 1) = x + y + 1 = ad(x, y) + 1$,

de onde se deduz que

$$f = p_1^1 : \begin{aligned} \mathbb{N}_0 &\to \mathbb{N}_0 \\ x &\mapsto x \end{aligned} \quad \text{e} \quad g : \begin{aligned} \mathbb{N}_0^3 &\to \mathbb{N}_0 \\ (x, y, z) &\mapsto z + 1 \end{aligned} \; .$$

Definição

As *funções recursivas primitivas* são as funções iniciais e todas aquelas que podem ser obtidas das funções iniciais pela aplicação de um número finito de vezes das operações de composição e de recursão primitiva.

Exemplo 4

A função ad, de adição em $\mathbb{N}_0$, é recursiva primitiva. De facto, como vimos no Exemplo 3, $ad = \operatorname{Rec}(f, g)$ é obtida por recursão primitiva das funções

$$f = p_1^1 : \begin{array}{ccc} \mathbb{N}_0 & \to & \mathbb{N}_0 \\ x & \mapsto & x \end{array} \quad \text{e} \quad g : \begin{array}{ccc} \mathbb{N}_0^3 & \to & \mathbb{N}_0 \\ (x, y, z) & \mapsto & z + 1 \end{array}.$$

Ora

- $f = p_1^1$ é uma função inicial;
- $g = s \circ p_3^3$ é a composição de duas funções iniciais.

Logo a adição em $\mathbb{N}_0$ é recursiva primitiva. Podemos decompô-la como

$$ad = \operatorname{Rec}(p_1^1, s \circ p_3^3).$$

Problema 3.2

1. Mostre que as seguintes funções são recursivas primitivas.

   a) $pred(x) = \begin{cases} 0 & \text{se } x = 0 \\ x - 1 & \text{se } x > 0 \end{cases}$ b) $sgn(x) = \begin{cases} 0 & \text{se } x = 0 \\ 1 & \text{se } x > 0 \end{cases}$

   c) $monus(x, y) = x \dot{-} y = \begin{cases} x - y & \text{se } x \geq y \\ 0 & \text{se } x < y \end{cases}$

   d) $max(x, y) = \begin{cases} x & \text{se } x \geq y \\ y & \text{se } x < y \end{cases}$ e) $dist(x, y) = \begin{cases} x - y & \text{se } x \geq y \\ y - x & \text{se } x < y \end{cases}$

2. Verifique quais das seguintes funções $f : \mathbb{N}_0 \to \mathbb{N}_0$ são totais calculando o seu domínio D_f.

   a) $f(x) = x/4$ b) $f(x) = x + 5$ c) $f(x) = x^2 - 9$
   d) $f(x) = (x - 3)^2$ e) $f(x) = x \dot{-} 8$

Resposta (sucinta)

1. a) $pred = Rec(zero^0, p_1^2)$. b) $sgn = Rec(zero^0, s \circ zero^2)$. c) $monus = Rec(p_1^1, pred \circ p_3^3)$.
   d) $max = ad \circ (monus, p_2^2)$. e) $dist = ad \circ (monus, monus \circ (p_2^2, p_1^2))$.

2. a) $D_f = 4\mathbb{N}_0$; f não é total. b) $D_f = \mathbb{N}_0$; f é total. c) $D_f = \mathbb{N}_0 \setminus \{0, 1, 2\}$; f não é total.
   d) $D_f = \mathbb{N}_0$; f é total. e) $D_f = \mathbb{N}_0$; f é total.

Teorema

Todas as funções recursivas primitivas são computáveis.

Demonstração: Basta notar que todas as funções iniciais são computáveis e que as funções obtidas de funções computáveis por composição ou recursão primitiva ainda são computáveis. □

Teorema

Todas as funções recursivas primitivas são funções totais.

Demonstração: Basta notar que as funções iniciais são totais e que as funções obtidas de funções totais por composição ou recursão primitiva ainda são totais. □

Dado que existem funções computáveis que não são totais, deduz-se que a classe das funções recursivas primitivas está propriamente contida na classe das funções computáveis. Pode-se provar mesmo o seguinte resultado.

Teorema

Existem funções totais computáveis que não são recursivas primitivas.

Exemplo 5

Seja $A : \mathbb{N}_0^2 \to \mathbb{N}_0$ a função definida por:

i) $A(0, y) = y + 1$;

ii) $A(x + 1, 0) = A(x, 1)$;

iii) $A(x + 1, y + 1) = A\big(x, A(x + 1, y)\big)$.

A função A é chamada função de Ackermann.

A função A é uma função *total computável*. Tem-se por exemplo,

$$\begin{aligned} A(1,1) &= A\big(0, A(1,0)\big) \quad \text{por } iii) \\ &= A(1,0)+1 \quad \text{por } i) \\ &= A(0,1)+1 \quad \text{por } ii) \\ &= 3 \quad \text{por } i). \end{aligned}$$

$$A(4,y) = \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{y+3 \text{ vezes}} -3.$$

TEOREMA

A função de Ackermann *não é recursiva primitiva*.

Problema 3.3

a) Determine $A(1,2)$ e $A(2,1)$.

b) Sabendo que $A(2,y) = 2y + 3$ para todo o $y \in \mathbb{N}_0$, prove que $A(3,y) = 2^{y+3} - 3$ para qualquer $y \in \mathbb{N}_0$.

c) Mostre que $A(4,y) = \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{y+3 \text{ vezes}} -3$ para qualquer $y \in \mathbb{N}_0$.

Definição (operador de minimização)

Se $f : \mathbb{N}_0^{k+1} \to \mathbb{N}_0$ é uma função, então a *minimização* (ilimitada) de f é a função (parcial)

$$\begin{array}{rccl} M_f : & \mathbb{N}_0^k & \to & \mathbb{N}_0 \\ & \vec{x} & \mapsto & \min\{y \in \mathbb{N}_0 : f(\vec{x}, y) = 0\}, \end{array}$$

onde $\vec{x}$ representa um k-uplo $\vec{x} = (x_1, \ldots, x_k)$ arbitrário.

Por vezes usa-se a notação

$$\mu y[f(\vec{x}, y) = 0] \text{ em vez de } M_f(\vec{x}).$$

Exemplo 6

Seja f a função ternária

$$f = p_1^3 : \begin{array}{ccc} \mathbb{N}_0^3 & \rightarrow & \mathbb{N}_0 \\ (x_1, x_2, x_3) & \mapsto & x_1 \end{array}$$

de projeção na primeira componente. A função M_f, de minimização de f, é dada por

$$M_f : \begin{array}{ccl} \mathbb{N}_0^2 & \rightarrow & \mathbb{N}_0 \\ (x_1, x_2) & \mapsto & \min\{y \in \mathbb{N}_0 : f(x_1, x_2, y) = 0\} \\ & & = \min\{y \in \mathbb{N}_0 : x_1 = 0\} \end{array}$$

Logo, a função M_f apenas está definida nos pontos (x_1, x_2) tais que $x_1 = 0$, tendo-se

$$M_f(0, x_2) = \min \mathbb{N}_0 = 0.$$

Exemplo 7

Consideremos agora a função ternária

$$g = p_3^3 : \begin{array}{rcl} \mathbb{N}_0^3 & \rightarrow & \mathbb{N}_0 \\ (x_1, x_2, x_3) & \mapsto & x_3 \end{array}$$

de projeção na terceira componente. Neste caso tem-se

$$M_g : \begin{array}{rcl} \mathbb{N}_0^2 & \rightarrow & \mathbb{N}_0 \\ (x_1, x_2) & \mapsto & \min\{y \in \mathbb{N}_0 : g(x_1, x_2, y) = 0\} \\ & & = \min\{y \in \mathbb{N}_0 : y = 0\} \\ & & = \min\{0\} \\ & & = 0 \end{array}$$

Ou seja,

$$M_g = zero^2.$$

Exemplo 8

Seja g a função

$$\begin{array}{rccl} g: & \mathbb{N}_0^2 & \to & \mathbb{N}_0 \\ & (x,y) & \mapsto & \begin{cases} 0 & \text{se } x = y^2 \\ 1 & \text{senão} \end{cases} \end{array}$$

A função M_g, de minimização de g, é a função

$$\begin{array}{rccl} M_g: & \mathbb{N}_0 & \to & \mathbb{N}_0 \\ & x & \mapsto & \min\{y \in \mathbb{N}_0 : g(x,y) = 0\} \\ & & & = \min\{y \in \mathbb{N}_0 : x = y^2\} \\ & & & = \begin{cases} \sqrt{x} & \text{se } x \text{ é um quadrado perfeito} \\ n.d. & \text{senão} \end{cases} \end{array}$$

Definição

Uma função diz-se *parcial μ-recursiva* (ou simplesmente *parcial recursiva*) se é uma função inicial ou pode ser obtida destas pela aplicação de um número finito de vezes das operações de composição, recursão primitiva e minimização.
Uma função parcial recursiva que seja total diz-se *recursiva*.

Teorema

Uma função $f : \mathbb{N}_0^k \to \mathbb{N}_0$ é parcial recursiva se e só se é computável.