

2. Algoritmos & computadores

DIM0320

2015.1

Sumário

- 1 Origem e história
- 2 Definições
- 3 Noções de lógica
- 4 Computadores
- 5 Linguagens e abstração

- 1 Origem e história
- 2 Definições
- 3 Noções de lógica
- 4 Computadores
- 5 Linguagens e abstração

Primeira ocorrência

1957

Entrada no Webster da palavra **algorithm**.

Antes

- **algorism** (em inglês) = "aritmética com números arábicos"
- **algarismo** (em português) \approx dígito

Origem

Um livro

- **Kitab al-jabr wa'l-muqabala** por
Abū 'Abd Allāh Muhammad ibn Mūsā al – Khwārizm (ca. 825),
- Livro sobre soluções de equações lineares e quadráticas.

Depois

Esquecimento

Essa origem foi esquecida durante o período do Renascimento. A palavra mudou para **algoritmo**, reconstruída erroneamente como

algoritmo = *alg*iros (dolorosos) *arithmo*s (números)...

Até 1950 ...

algoritmo \approx algoritmo de Euclides.

- 1 Origem e história
- 2 Definições**
- 3 Noções de lógica
- 4 Computadores
- 5 Linguagens e abstração

Sentido

Atenção!

Algoritmo \neq logaritmo.

Definição (Definição geral)

O sentido geral é o mesmo de receita, processo, método, técnica ... e isso caracteriza uma sequência finita de operações (ou regras) para resolver um problema.

Definição detalhada

Definição (Knuth)

De acordo com D. Knuth [Knu97], um algoritmo tem 5 características.

- 1 Ser finito
- 2 Ser bem definido
- 3 Ter entradas
- 4 Ter uma saída
- 5 Eficácia

Ser finito

Um algoritmo deve terminar depois de um número finito de passos. Caso contrário, é chamado de **método computacional**.

Ser bem definido

Cada passo dever ser sem ambiguidade e as definições das ações devem ser rigorosas.

- usar **linguagens naturais** causa problemas de ambiguidade
- é melhor usar linguagens de programação (com vocabulário restrito) com uma semântica bem definida. O **Portugol** é uma tal linguagem.
- A expressão de um método computacional é chamada de **programa**.

Ter entradas

Definição (Entrada)

Uma entrada é um conjunto de dados fornecido antes ou durante a execução.

Um algoritmo tem um número de entradas ≥ 0

Ter uma saída

Definição (Saída)

Uma saída é um conjunto de dados produzidos pela execução do algoritmo.

Um algoritmo tem um número de saídas ≥ 1

Ser eficaz

Esperamos descrever algoritmos eficientes/eficazes. As operações podem ser entendidas ou feitas em tempo finito por uma pessoa com caneta / papel (pode ser difícil e longo).

O Algoritmo

História

- Detalhado por Euclides, ca. 300 a.C. nos livros VII e X dos **Elementos**.
- Conhecido também na China, durante a dinastia Han (200 a.C – 200 d.C.).
- Esse algoritmo calcula o **máximo divisor comum**.

Algoritmo de Euclides

[Encontrar resto]

E1. Dividir m por n . Seja r o resto dessa divisão ($0 \leq r \leq n$)

[É zero ?]

E2. Se $r = 0$, terminar. A resposta é o valor de n .

[Reduzir]

E3. $m \leftarrow n$, $n \leftarrow r$, voltar para E1.

Em Portugal

```
algoritmo "MCD"  
var m, n, r: inteiro  
inicio  
    leia(m)  
    leia(n)  
    // hip  $m > n$   
    r <- n //  $<> 0$   
    n <- m  
    enquanto r  $<> 0$  faça  
        m <- n  
        n <- r  
        r <- m % n  
    fimenquanto  
    escreva(n)  
finalgoritmo
```

Cálculo de área

Objetivo

Calcular a área 'A' de um quadrado 'Q' de lado 'l'.

Algoritmo

[Solicitar]

A1. Solicitar o comprimento l do lado do quadrado Q .

[Multiplicar]

A2. Multiplicar l por l .

[Escrever]

A3. Escrever o resultado de A2 numa folha de papel.

Em Portugal

```
algoritmo "Area"  
var l : inteiro  
inicio  
    escreval("Entre com o comprimento do lado")  
    leia(l)  
    escreva(l * l)  
finalgoritmo
```

Algoritmo como processamento

Processamento

Um algoritmo é um **processamento** (uma transformação) de dados.



Solução e algoritmos

- A solução do problema **não é** o algoritmo.
- Algoritmo = procedimento para encontrar a solução.
- Algoritmos calculam apenas soluções de ocorrências (i.e. instâncias de entradas) de problemas computacionais.

Correção de algoritmos

Definição (Correção)

Um algoritmo **correto** para o problema P tem duas características:

- ① Ele termina.
- ② Ele produz uma resposta correta para **todas** ocorrências de P.

Nesse modelo, o computador executa fielmente as instruções do algoritmo. Erros de computadores são tão raras que esse modelo é perto da realidade.

Determinar a correção

A área de **verificação de software** estuda a correção do software em relação às especificações dele.

Representações de algoritmos

Descrição

Algoritmos podem ser descritos por

- linguagens; ou
- representações gráficas (**fluxogramas**, ...).

Porque usar linguagens?

Uma linguagem algorítmica devem ser

- 1 restrita (vocabulário pequeno)
- 2 bem definida (sem ambiguidade)
- 3 suficientemente expressiva

Como resolver um problema?

Um método possível

- 1 Identificar E/S
- 2 Determinar (em português) a forma de solucionar o problema computacional
- 3 Descrever **2** como uma sequência finita de passos
- 4 Traduzir para uma linguagem

Como para a prova de teoremas, o pensamento lógico é muito importante

- 1 Origem e história
- 2 Definições
- 3 Noções de lógica**
- 4 Computadores
- 5 Linguagens e abstração

Origem da lógica (a.C.)

Definição

Pensamento estruturado e sistematizado (através de regras de dedução por ex.)

As origens

Contribuições notáveis por

- China (Mozi/Mo-tzu séc.5), estudo proibido
- Índia (Gautama séc.6 , Panini séc.5, Chanaky – séc.4), independentemente
- Grécia (Plato, Aristóteles, século 4 a.C.)

Modernidade

Lógica moderna descende da lógica aristotélica (clássica)

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.
- 2 Todo cavalo é um mamífero.

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.
- 2 Todo cavalo é um mamífero.
- 3 (\Rightarrow) Logo, todo cavalo é um?

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.
- 2 Todo cavalo é um mamífero.
- 3 (\Rightarrow) Logo, todo cavalo é um?
 - ▶ animal

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.
 - 2 Todo cavalo é um mamífero.
 - 3 (\Rightarrow) Logo, todo cavalo é um?
 - ▶ animal
-
- 1 Neymar é jogador do time de futebol brasileiro.

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.
 - 2 Todo cavalo é um mamífero.
 - 3 (\Rightarrow) Logo, todo cavalo é um?
 - ▶ animal
-
- 1 Neymar é jogador do time de futebol brasileiro.
 - 2 O time brasileiro perdeu . . .

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.
 - 2 Todo cavalo é um mamífero.
 - 3 (\Rightarrow) Logo, todo cavalo é um?
 - ▶ animal
-
- 1 Neymar é jogador do time de futebol brasileiro.
 - 2 O time brasileiro perdeu . . .
 - 3 1-7

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.
 - 2 Todo cavalo é um mamífero.
 - 3 (\Rightarrow) Logo, todo cavalo é um?
 - ▶ animal
-
- 1 Neymar é jogador do time de futebol brasileiro.
 - 2 O time brasileiro perdeu . . .
 - 3 1-7
 - 4 (\Rightarrow) Logo, Neymar fez o que?

O que é lógica ?

- A lógica estuda e ensina a colocar ordem no pensamento
- Sinônimos: coerência, racionalidade
- Uma de suas preocupações é determinar quais operações são válidas e quais não são, avaliando a corretude do pensamento

Exemplo

- 1 Todo mamífero é um animal.
- 2 Todo cavalo é um mamífero.
- 3 (\Rightarrow) Logo, todo cavalo é um?
 - ▶ animal

- 1 Neymar é jogador do time de futebol brasileiro.
- 2 O time brasileiro perdeu . . .
- 3 1-7
- 4 (\Rightarrow) Logo, Neymar fez o que?
 - ▶ perdeu

Primeiros problemas

Aquiles e a tartaruga

- Corrida entre Aquiles e a tartaruga
- A tartaruga recebe uma vantagem: começa na frente
- Aquiles vai alcançar a tartaruga ?

Paradoxo de Epiménides

Esta afirmação é falsa.

- 1 Origem e história
- 2 Definições
- 3 Noções de lógica
- 4 Computadores**
- 5 Linguagens e abstração

Primeira definição

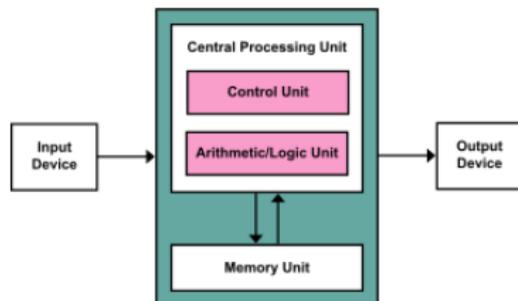
Definição (Computador)

Instrumento capaz de

- ① realizar cálculos
- ② tomar decisões lógicas

Um computador é **bilhões de vezes** mais rápido que um ser humano.
Mas ele deve ser programado.

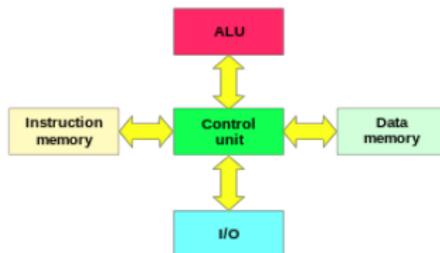
Arquitetura von Neumann



Divisão em unidades lógicas

- 1 Entrada
- 2 Saída
- 3 Memória
- 4 Unidade aritmética e lógica (ALU)
- 5 Unidade central de processamento (CPU)
- 6 Unidade de armazenamento

Arquitetura Harvard



- **Dados e programas separados**
- Acesso simultâneo ao 2 canais

Entradas

Papel

Obter informações (programas, dados) de dispositivos de entrada.
As informações são usadas por outras unidades.

Dispositivos de entrada

- teclado
- discos
- rede

Saídas

Papel

Mandar informações/dados.

Pega as informações processadas/produzidas pelo computador e coloca-as nos dispositivos de saída.

Dispositivos de saída

- impressão
- tela
- rede
- discos

Memória

Papel

Armazenar as informações, lidas pelos dispositivos de entrada ou produzidos antes para os dispositivos de saída.

Memória

Falamos aqui da memória de acesso rápido (RAM, cache) e de capacidade reduzida.

ALU / UAL

Papel

- Fabricar os dados,
- realizar cálculos,
- comparar 2 elementos da memória para tomar decisões.

Papel

Administrar o computador, coordenar, supervisionar operações de E/S, da memória, da ALU et da unidade de armazenamento.

- Informar entradas que informações devem ser lidas da memória.
- Informar a ALU que ela precisa usar dados da memória.
- Informa saídas que tem que mandar dados da memória.

Papel

- Unidade de armazenamento de longo prazo e alta capacidade.
- Armazena por exemplo programas ou dados não ativamente usados pelas outras unidades, até o próximo uso.

Ordem de grandeza e computadores

Elemento	Latência	Comparação
Cache L1	0.5 ns	batida coração
Predição de ramo errada	5 ns	bocejo
Cache L2	7 ns	bocejão
Sincronização dado	25 ns	fazer café
RAM	100 ns	lavar dentes
Comprimir 1K bytes	3 μ s	novela
Enviar 2KB (rede 1GBps)	20 μ s	almoço \rightarrow fim do dia
SSD	150 μ s	fim de semana
Ler 1 MB da RAM	250 μ s	fim de semana + feriado
Ida/volta num datacenter	0.5 ms	semana de férias
Ler 1 MB de SSD	1 ms	natal-reis magos
Busca num HD	10 ms	semestre letivo
Ler 1 MB do HD	20 ms	gravidez
Enviar pacote USA \rightarrow Europa \rightarrow USA	150 ms	bacharelado

- Leituras sequenciais

- 1 Origem e história
- 2 Definições
- 3 Noções de lógica
- 4 Computadores
- 5 Linguagens e abstração**

Linguagens: a situação

Ser humano

Eu quero escrever "Hello, World!".

Prático (para nós) mais ambíguo, impreciso. E geralmente impossível traduzi-lo para o computador.

Computador

O computador entende só 0 e 1 (binário).

Simples, preciso mas incômodo (para nós).

Humano

Código

Eu quero escrever "Hello, World!".

Binário (máquina)

Código (Extrato)

```
01111111 01000101 01001100 01000110 00000010 00000001 00000001 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000010 00000000 00111110 00000000 00000001 00000000 00000000 00000000
10000000 00011101 01000000 00000000 00000000 00000000 00000000 00000000
01000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
10110000 11101010 00000001 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 01000000 00000000 00111000 00000000
00001000 00000000 01000000 00000000 00100010 00000000 00011111 00000000
00000110 00000000 00000000 00000000 00000101 00000000 00000000 00000000
01000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
01000000 00000000 01000000 00000000 00000000 00000000 00000000 00000000
01000000 00000000 01000000 00000000 00000000 00000000 00000000 00000000
11000000 00000001 00000000 00000000 00000000 00000000 ...
```

Assembly

Código (extrato)

```
.Ltmp4:
    .cfi_def_cfa_register %rbp
    subq    $16, %rsp
    leaq    .L.str, %rdi
    movb    $0, %al
    callq   printf
    movl    $0, %ecx
    movl    %eax, -4(%rbp)           # 4-byte Spill
    movl    %ecx, %eax
    addq    $16, %rsp
    popq    %rbp
    ret

.L.str:
    .asciz  "Hello, World!\n"
    .size  .L.str, 15
```

Linguagem baixo nível

Código (extrato)

```
; ModuleID = 'hello.c'

@.str = private unnamed_addr constant [15 x i8] c"Hello, World!\0A\00", ali

; Function Attrs: nounwind uwtable
define i32 @main() #0 {
    %1 = call i32 @i8*, ...)*
        @printf(i8* getelementptr inbounds ([15 x i8]* @.str,
            i32 0, i32 0))
    ret i32 0
}

declare i32 @printf(i8*, ...) #1
....
```

Linguagem C

Código

```
#include <stdio.h>  
main () {  
    printf("Hello, World!\n");  
}
```

Portugol

Código

```
algoritmo "hello"  
inicio  
    escreval("Hello, World!")  
finalgoritmo
```

Referências



Donald E. Knuth, *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*, Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1997.

Perguntas ?



<http://dimap.ufrn.br/~richard/dim0321>