Introdução à Ciência da Computação

Módulo 2

Luis Retondaro

CEFET-RJ
Campus Petrópolis

August 15, 2023

Sumário

Algoritmo e Programa

Algoritmo / Programa

Devemos distinguir claramente dois aspectos: estático e dinâmico.

- O aspecto estático de um algoritmo consiste no texto contendo as instruções a serem executadas. Esse texto representa concretamente o algoritmo e tem um caráter atemporal, estático.
- O aspecto dinâmico de um algoritmo está além do texto. Está nos efeitos da execução de cada instrução no tempo, dado um conjunto de valores iniciais.

Aspectos Estático e Dinâmico

Estruturas

- De fato, a grande dificuldade na criação e no entendimento de algoritmos está no problema do relacionamento desses aspectos, ou seja, como entender e visualizar as estruturas dinâmicas das possíveis execuções do algoritmo a partir da estrutura estática do seu texto.
- Para facilitar, limitamos a quantidade de estruturas de controle em um algoritmo em poucas **sequências simples**, **alternativas** e **repetições**. Uma ação é um evento que ocorre num período de tempo finito, estabelecendo um efeito intencionado e bem definido.

Exemplos:

- **“caminhar até a próxima esquina”**
- **“descascar batatas”**

Evento / Processo

Estado de um objeto

- Outra coisa importante a considerar é que o efeito de uma ação não pode ser imprevisível.
- O **estado** de um objeto no tempo são as propriedades que são relevantes para nós na situação considerada.
- Por exemplo: **batatas com casca ou descascadas**.
- No caso de um programa, podemos considerar um determinado valor num certo instante da execução.

Processo

- Quando consideramos um evento como uma sequência de ações, cujo efeito acumulado é igual ao efeito do evento total, falamos de um **processo sequencial**, ou simplesmente de um **processo**.

Padrão de comportamento

Evento

- Para descrever um evento, inicialmente, usamos a forma de relato de um observador.
- Por exemplo, para o evento **“a cozinheira descasca as batatas para o jantar”**, as ações podem ser separadas por **“;”** e descritas na seguinte sequência:

EV1 {
“traz a cesta com batatas para a cozinha” ;
“pega a panela no armário” ;
“descasca as batatas”.

Relatos iguais de eventos distintos

- Nesta situação reconhecemos um **padrão de comportamento**

Padrão de comportamento

Efeitos

- em todo evento podemos reconhecer um padrão de comportamento, fazendo abstração dos possivelmente diferentes estados iniciais e efeitos.
- Inversamente, cada vez que o padrão de comportamento é "seguido" o evento ocorre.
- O efeito de um evento está totalmente determinado pelo padrão de comportamento e pelo estado inicial.

Por exemplo:

- Nas diferentes execuções da operação n^2, para diferentes valores de n, podemos reconhecer um mesmo padrão de comportamento, e a ação de cada um dos eventos é "**elevar um número ao quadrado**".
- Neste caso, o estado inicial é dado pelo valor de n, e o efeito, pelo valor obtido multiplicando-se esse valor por si mesmo.


VIDA DE PROGRAMADOR .COM.BR
real historia;
string sender;
sender = "Fabrício Olmo Aride";
BY NC SA
#1421
E AÍ, JÁ RESOLVEU O PROBLEMA?
AINDA NÃO...
O PROBLEMA É QUE O SISTEMA TEM MAIS LÓGICA DO QUE VOCÊ
O QUE ESTÁ ACONTECENDO?
ENTÃO, A LÓGICA ESTÁ CORRETA...
O PROBLEMA É QUE O SISTEMA TEM MAIS LÓGICA DO QUE VOCÊ
MAS O SISTEMA NÃO FAZ O QUE EU QUERO...
PLOFT!

Lógica formal

Definição resumida

- Simplificadamente, a lógica pode ser dividida em lógica formal e lógica material.
- A lógica formal não depende das verdades de cada premissa, mas da relação destas com a conclusão, afim de determinar se elas podem sustentar a conclusão.
- Nesse caso, para a lógica formal, todas as premissas são verdadeiras.
- De forma muito resumida, podemos dizer que a lógica se preocupa com a formatação de um raciocínio, ou com a maneira que uma ideia pode ser expressa, a fim de possibilitar uma conclusão sobre determinado pensamento em busca da verdade.

Conclusão lógica

Argumentação

- Os dois tipos de argumentação usados na lógica formal são: a dedução e a indução.
- O método lógico dedutivo parte de fatos ou eventos universais ou mais genéricos, para concluir os mais particulares.
- O método indutivo analisa os fatos ou eventos particulares ou específicos para concluir uma hipótese mais geral, abrangente, universal, genérica.

Dedução x indução lógica

Dedução lógica ou método dedutivo

- “Todos os humanos são mortais.”
- “Todos os petropolitanos são humanos.”
- “Todos os petropolitanos são mortais.”

Indução lógica ou método indutivo

- “A macieira é uma fruteira.”
- “O limoeiro é uma fruteira.”
- “A macieira e o limoeiro são árvores.”
- “As árvores são fruteiras.”

Método dedutivo

Atenção

- Note que, apesar da indução ser um processo natural, suas conclusões podem ser perigosas, pois generalizar premissas verdadeiras podem levar a uma falsa conclusão.
- Nesse caso, não se pode dizer que a conclusão do argumento é uma verdade.
- Já que os argumentos dedutivos são considerados válidos pela forma lógica e não pelo conteúdo dos seus enunciados, é neste método que se baseia a lógica de construção de algoritmos.

Principais proposições lógicas

Conjunção - e / and

- Ambas as sentenças necessitam ser verdadeiras para a saída ser verdadeira
- Representado pelo produto $p \cdot q$

P	Q	P.Q
0	0	0
0	1	0
1	0	0
1	1	1

Principais proposições lógicas

Disjunção - ou / or

- Basta que apenas uma das sentenças seja verdadeira para a saída ser verdadeira
- Representado pela soma $p + q$

P	Q	$P + Q$
0	0	0
0	1	1
1	0	1
1	1	1


VIDADEPROGRAMADOR
.COM.BR
/* SUGESTÃO ENVIADA POR JOSEPH F */
#25
CARA, EU NUNCA VOU ENTENDER AS MULHERES
NINGUÉM VAI... MAS POR QUE VOCÊ DIZ ISSO?
ROOT
MINHA ESPOSA ME PEDIU PARA IR AO MERCADO E DISSE: "TRAGA 6 OVOS. SE TIVER BATATAS, TRAGA 9!"
ROOT
TINHA BATATAS. DAÍ ELA FICOU BRAVA PORQUE EU LEVEI SÓ OS 9 OVOS...
VAI ENTENDER...
ROOT

Controle de fluxo

Sequência de comandos

- O conceito que relaciona o aspecto estático de um algoritmo com o seu aspecto dinâmico, é o de **controle de fluxo**.
- Ele determina em cada passo da execução qual é o próximo comando a ser executado.
- A ordem de execução dos comandos, geralmente não é a mesma ordem descrita no texto estático.
- Compreender a lógica de um algoritmo significa visualizar os processos (sequência de comandos) que serão executados, dependendo do fluxo de controle.
- Vejamos a seguir, um exemplo que utiliza apenas as estruturas básicas de controle: **sequência** e **alternativa**:

Controle de fluxo

Exemplo ilustrativo

```
se a > 15                                   { a > 15   }
  então
       x ← x * 4;
       y ← y + 3;
  senão se a > 10                           { 10 < a ⩽ 15   }
            então
                 x ← x * 3;
                 y ← y + 2;
            senão
                 x ← 0;
                 y ← 0;
       fim se;
fim se;
```

Típico programa em BASIC

Conhecendo ou não o efeito de cada comando, tente reconhecer o fluxo

```
10  REM RESOLVE EQUACAO DO SEGUNDO GRAU
20  READ A,B,C
30  IF A=0 THEN GOTO 400
40  LET D=B*B-4*A*C
50  IF D<0 THEN GOTO 420
60  PRINT "SOLUCAO"
70  IF D=0 THEN GOTO 200
80  PRINT "PRIMEIRA SOLUCAO",(-B+SQR(D))/(2*A)
90  PRINT "SEGUNDA SOLUCAO",(-B-SQR(D))/(2*A)
100  GOTO 20
200 PRINT "SOLUCAO UNICA",(-B)/(2*A)
300 GOTO 20
400 PRINT "A DEVE SER DIFERENTE DE ZERO"
410 GOTO 20
420 PRINT "NAO HA SOLUCOES REAIS"
430 GOTO 20
490 DATA 10,20,1241,123,22,-1
500 END
```

Típico programa "Hello, world" em COBOL

Hello, world

```
*****************************
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO.
ENVIRONMENT DIVISION.
DATA DIVISION.
PROCEDURE DIVISION.
MAIN SECTION.
DISPLAY "Hello World!"
STOP RUN.
*****************************
```

Típico programa "Hello, world" em Assembly

Hello, world

```
; Hello World for Intel Assembler (MSDOS)

mov ax,cs
mov ds,ax
mov ah,9
mov dx, offset Hello
int 21h
xor ax,ax
int 21h

Hello:
  db "Hello World!",13,10,"$"
```

Típico programa "Hello, world" em C

Hello, world

```c
#include <stdio.h>

int main()
{
  printf("Hello World!\n");

  return 0;
}
```

Típico programa "Hello, world" em Python

Hello, world

```
print("Hello World")
```

Fluxogramas

Fluxogramas

Esquema semântico

- Fluxograma é um tipo de diagrama esquemático que representa bem o aspecto dinâmico de um algoritmo.
- O fluxo de execução, bem como as estruturas de controle são evidenciadas e é possível compreender a semântica da solução.
- Em toda linguagem, as sentenças construídas envolvem dois aspectos: a sintaxe e semântica.
- A sintaxe tem a ver com a forma e a semântica com o conteúdo.

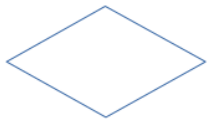
Fluxogramas

Fluxogramas

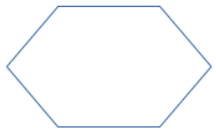
As notações usadas em um fluxograma são:


Marca o início e o fim do algoritmo


Denota um processo/comando

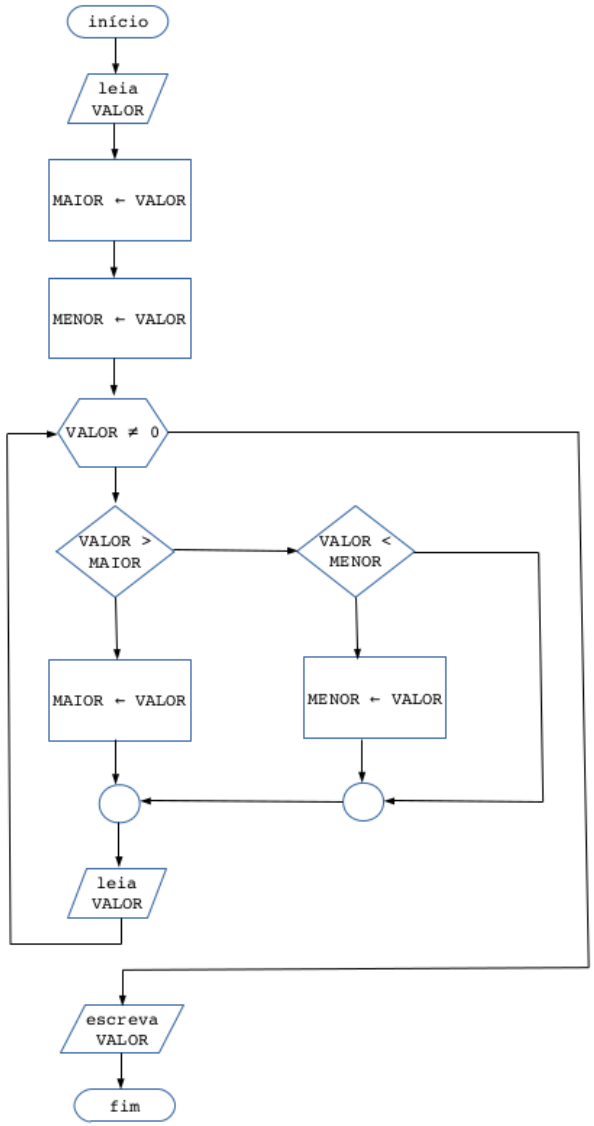

Representa um desvio condicional


Usado como conector. Geralmente, marca o fim de um bloco


Representa um bloco de repetição, com condição inicial.


Entrada e Saída

Fluxogramas



Pseudo-código (Portugol)

Estruturas básicas

- O texto do programa é chamado de **código-fonte**.
- Simplificadamente, chamamos apenas de **código**.
- O Pseudo-código é uma abstração da linguagem de programação a ser utilizada

Identificador

- Elemento básico da linguagem
- Representa o nome de uma estrutura ou uma variável

Sintaxe:

- **Válidos:** A15, IDADE, Menor, x2, First_name
- **Inválidos:** 20x, Maior peso, %3k

Estruturas básicas

Variável

- Identidade de um espaço de armazenamento de conteúdo/valor
- Se precisamos armazenar um determinado valor X, podemos alocar um espaço na "memória" denominado X.
- o símbolo "$\leftarrow$" normalmente é substituído por "$=$" nas linguagens de programação

Sintaxe:

- $x \leftarrow y + 4$
- $soma \leftarrow soma + 1$
- $S \leftarrow 15$

Exemplo de Pseudo-código

Especificar um algoritmo para somar o valor de três variáveis e calcular a média desses valores.

- **Variáveis:** A1, A2, A3
- Soma = A1 + A2 + A3
- Média = Soma / 3

No exemplo acima há...

- 3 variáveis de entrada
- uma variável auxiliar (etapa de processamento)
- e uma variável de saída

Tipos de dados básicos

Inteiros

- Assumem somente valores inteiros
- **inteiro: X, Y;**
- $x = 5$
- $y = -4$

Real (ou ponto flutuante)

- Assumem qualquer valor de um número real
- **real: a, b, c;**
- $a = 1.25$
- $b = -3.78$
- $c = 8$

Tipos de dados básicos

Caracteres (ou strings)

- Assumem quaisquer dígitos alfanuméricos, inclusive símbolos
- São expressos entre aspas
- **Caracter: Nome, Tipo;**
- $Nome = ''Maria''$
- $Tipo = ''\%''$

Lógico (ou booleano)

- Assumem valores **FALSO** ou **VERDADEIRO** como resultados de expressões lógicas
- **Logico: Contratado, Finalizou**
- $Contratado = true$
- $Finalizou = false$

Outros tipos de dados e estruturas

ATENÇÃO

Há diversos outros tipos de dados especiais importantes que serão apresentados oportunamente.

Operadores

Operadores Aritméticos

- $+$ (soma)
- $-$ (subtração)
- $*$ (multiplicação)
- $/$ (divisão)
- *mod* (resto da divisão)
- *raiz* (sqrt - raiz quadrada)
- $**$ (pow - potenciação)

Exemplo:

$x = (b * *3) + 4$

Operadores

Operadores Lógicos

- **e** (&& - conjunção)
- **ou** (|| - subtração)
- **não** (! - negação)

Exemplo

$a = !(TERMINADO \,||\, cadastrado)$

Operadores

Operadores Relacionais

- **=** (== igualdade)
- **<>** (!= desigualdade)
- **>=** (maior ou igual)
- **<=** (menor ou igual)
- **>** (maior)
- **<** (menor)

Exemplo

$h = (a <= 10) \,||\, (b \,!= 5)$

Exercícios

Supondo as variáveis A, B e C

$A = true$ (verdadeiro), $B = False$ e $C = 10$, qual o resultado das seguintes expressões?

- $A \,||\, B$
- $A \,\&\&\, B$
- $(B \,||\, C >= 10)$
- $(A \,\&\&\, (C \bmod 2 == 0)) \,||\, B$

Processamento de Entrada e Saída (I/O)

Entrada

- Utilizado para receber valores de entrada e armazenar em uma variável
- **leia(<variável>)**

Saída

- Utilizado para "imprimir" um conteúdo na tela ou em um arquivo
- **escreva(<variável>)**

Estrutura de decisão

Alternativa ou desvio condicional

- Estruturas de códigos **NÃO** sequenciais

Sintaxe

- **se** <condição> **então**
  <ação>
  fimse
- **se** <condição> **então**
  <ação 1>
  senão
  <ação 2>

Laços de repetição

Loop

- Permitem executar instruções um certo número de vezes (iterações).
- Tipicamente, dois tipos: teste no início ou no final

Enquanto / while

enquanto <condição> **faça**
<ação>
fimenquanto

Repita / repeat

repita
<ação>
até <condição>

Laços de repetição

Para / for

para <variável auxiliar = valor inicial> **até** <valor final> **faça**
<ação>
fimpara

Exercícios

Construa os seguinte algoritmos:

- Um algoritmo para gerar e imprimir N termos da série de Fibonacci:
  $1, 1, 2, 3, 5, 8, 13, 21...$
  N >= 2 e deve ser informado pelo usuário.
- Construir um algoritmo para calcular a média de um conjunto de valores inteiros e positivos fornecidos pelo usuário. Como **flag**, pode-se assumir o valor -1.
  Expressar o algoritmo em Portugol e em Fluxograma.
- Escrever um algoritmo para calcular o fatorial de um número *N* fornecido pelo usuário. Expressar o algoritmo em Portugol e em Fluxograma.

Técnicas

A linguagem de programação

Técnicas de codificação

MANUAIS

- Refinamentos sucessivos
- Depuração manual ("chinês")
- Comentários e banners
- Arquivo tipo READ.ME

Técnicas de codificação

AUTOMÁTICAS

- Escopo de variáveis
- Organização em arquivos e diretórios
- Confecção de bibliotecas
- Uso de API (*applicaton Program Interface*)
- Sistema de versionamento
- Depuração automática
- Hypertexto
- Estruturas de dados ótimas

Paradigmas de programação

Paradigmas

- Programação estruturada/procedural
- Programação orientada a objetos
- Programação Funcional
- Programação Lógica

Tipos

- Compartilhamento de estados
- Programação imperativa
- Programação declarativa