

Introdução à Computação

Nelson S. dos Santos

Núcleo de Ciência de Dados e Computacional em Economia e Finanças
(e-CompFin)

Faculdade de Ciências Econômicas
Universidade Federal do Rio Grande do Sul

1 de Novembro de 2024

Sumário

- 1 Introdução
- 2 Problemas e soluções
- 3 O que é computar
- 4 Modelos Abstratos de Computador e Computabilidade
- 5 Linguagens de computador
- 6 Programação sequencial
- 7 Referências

Introdução

Problema

Como resolver problemas cuja solução exija um número extremamente elevado de operações, isto é, problemas trabalhosos demais para serem resolvidos manualmente?

Importância

- Na nossa vida prática, em grande parte das vezes, enfrentamos problemas cuja solução é extremamente trabalhosa, ainda que não seja difícil.
- Por exemplo, em matemática, inverter uma matriz de dimensão 15×15 é trabalhoso, mas não é difícil.
- Igualmente, coletar dados para uma pesquisa e os armazenar e processar é um trabalho muito extenuante para ser feito manualmente.

- Outro exemplo seria criar um cadastro de clientes para uma empresa de médio porte.
- Enfim, em diversos casos, temos problemas complexos (isto é, compostos por uma grande número de partes) cuja solução exijam a execução de um enorme número de operações.

Solução

Criar uma máquina que receba nossas instruções sobre como resolver nosso problema e as execute em nosso lugar. Tal máquina é chamada de **computador**.

Computador: definição

Um computador é uma máquina capaz de receber instruções e executá-las criada com a finalidade de **resolver problemas** cuja solução exija um número extremamente elevado de operações.

Problemas e soluções

Problemas

Problemas dividem-se em:

- **simples** - são aqueles que não podem ser divididos em múltiplas partes, ou
- **complexos** - são aqueles formados por múltiplas partes.

Soluções

- Problemas simples podem ser resolvidos com apenas uma única operação.
- Problemas complexos exigem múltiplas operações para se obter sua solução.

Algoritmo

Uma sequência lógica e finita de operações para resolver um problema é chamada de **algoritmo**.

- Problemas simples possuem soluções com algoritmos simples (isto é, formado por uma única operação).
- Problemas complexos exigem algoritmos complexos (formados por múltiplas operações).

Algoritmo e cálculos aritméticos

- Os primeiros algoritmos foram criados para resolver problemas de cálculo, sendo o algoritmo de Euclides para determinação do M.D.C entre dois números um dos primeiros e mais famosos conhecidos.
- Algoritmos podem ser representados pela regra de definição de uma função cujo domínio é o conjunto de dados de entrada e o contradomínio é conjunto de dados de saída.

Computabilidade

Diz-se que um problema é computável se pode ser resolvido por meio de um conjunto finito de passos (algoritmo).

Complexidade de algoritmos

- A complexidade de um algoritmo é o principal determinante da velocidade de solução de um problema.
- A complexidade de algoritmos é medida em termos do número de operações de adição (multiplicação e exponenciação) e subtração (divisão) necessárias para a solução do problema.
- Utiliza-se uma medida denominada **O** para avaliar a complexidade de algoritmos. Veja, por exemplo, [aqui](#).

O que é computar

O que é computar

- Computar é calcular.
- Sim... Mas o que é calcular?
- Calcular é operar sobre os elementos de um conjunto.
- Tá... Mas o que é operar sobre os elementos de um conjunto?

- Operar sobre os elementos de um conjunto é aplicar uma função aos elementos do conjunto onde ela está definida.
- Humm... Como assim?
- Veja o exemplo.

Exemplo

Considere o conjunto \mathbb{R} dos números reais e a função $f : \mathbb{R} \rightarrow \mathbb{R}$ definida por $f(x) = 2.x$. *Calcular* o valor da função no ponto $a \in \text{dom}(f)$ é aplicar a definição da função ao ponto a , ou seja, substituir o valor a na regra de definição da função. Assim:

$$f(a) = 2.a$$

Em particular, se $a = 2$, temos que:

$$f(2) = 2.2 = 4$$

A regra de definição da função DEFINE a sequência de passos que deve ser seguida para calcular o valor da função. Esta regra materializa a noção de **algoritmo**.

Introdução

Problemas e soluções

O que é computar

Modelos Abstratos de Computador e Computabilidade

Linguagens de computador

Programação sequencial

Referências

Modelos Abstratos de Computador e Computabilidade

O computador

Um computador é uma máquina capaz de receber instruções e executá-las criada com a finalidade de **resolver problemas** cuja solução exija um número extremamente elevado de operações. No mundo concreto tal máquina tem as seguintes propriedades:

- É um dispositivo físico (mecânico, eletrônico, quântico etc) capaz de executar algoritmos, isto é, calcular valores de funções.
- O conjunto de sinais usados para comunicar ao computador o algoritmo a ser executado formam sua linguagem de máquina.

- Os computadores atuais são, na sua esmagadora maioria, dispositivos eletrônicos, o que significa que sua linguagem é formada por sinais elétricos.
- Um programa é uma tabela que associa a cada instrução do algoritmo uma instrução em linguagem de máquina.

Modelo Abstrato de Computador

- Como qualquer dispositivo complexo criado por seres humanos, o computador antes de ser construído é projetado por um engenheiro. A formalização matemática da concepção sobre a qual se define o projeto de um computador é chamado de modelo (ou máquina) abstrata.
- Um modelo abstrato de Computador é, pois, uma proposta para organização dos dispositivos que comporão o computador projetado.

Os modelos abstratos de Computador mais importantes são:

- Máquina de Turing, e
- Máquina de Von Neumann.

Máquina de Turing

- Dispositivo de entrada de dados (cabeçote de leitura de dados, teclado, mouse etc)
- Dispositivo de processamento de dados com armazenamento de dados (unidade central de Processamento com registrador de estados)
- Um **programa** que consiste em uma tabela que diz ao computador a instrução a ser executada a cada conjunto de dados (estado).

- Dispositivo de escrita de dados (cabeçote de escrita de dados)
- Veja mais, clicando [aqui](#).

Em resumo, o modelo de processamento de uma máquina de Turing consiste em:

- Entrada (ou leitura) de dados
- Processamento de dados por meio de um programa executado em uma unidade central de processamento com registrador de estados (memória)
- Saída (ou escrita) de dados.

Observe que este modelo de processamento é equivalente ao modelo de computação baseado em funções apresentado anteriormente o qual é denominado de **Cálculo Lambda**.

Conjectura de Church-Turing

A [Conjectura de Church-Turing](#) se constitui na hipótese de que um problema é computável se, e somente se, pode ser resolvido por meio de programa executado por uma Máquina de Turing.

Máquina de von Neumann

- É uma variação sobre a máquina de Turing onde a sequência de instruções que forma o algoritmo (isto é, o programa) é armazenada junto com os dados num dispositivo denominado memória.
- A máquina de von Neumann é formada pelos seguintes componentes:
 - dispositivos de entrada
 - dispositivo de saída
 - unidade central de processamento (composta de unidade lógica e aritmética e unidade de controle)
 - memória capaz de armazenar dados e o programa
- Veja mais informações em [Arquitetura de von Neumann](#).

A figura a seguir, extraída da [Wikipedia](#), mostra a arquitetura da máquina de von Neumann:

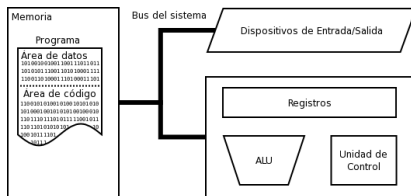


Figura 1: Arquitetura de von Neumann

Linguagens de computador

Comunicação com computadores

- Para mandar o computador executar as operações que desejamos, precisamos de uma linguagem que o computador seja capaz de entender.
- Mas, exatamente, o que é uma linguagem?
- Uma linguagem é um conjunto de sinais (chamado de léxico) unidos de modo a respeitar um conjunto de regras lógicas (denominadas de sintaxe) usadas para dissertar sobre um conjunto de objetos ou ideias (semântica).

- Dispositivos eletrônicos se caracterizam por responder apenas à presença ou ausência de sinais elétricos (ou sua voltagem/amperagem).
- Ora, os sinais elétricos obedecem às leis do eletromagnetismo, que são formuladas matematicamente (isto é, obedecem à lógica matemática).

Linguagem de máquina

- Usando estes fatos, podemos, então, formar uma **linguagem** com os seguintes elementos: sinais elétricos como **léxico**, a lógica matemática como a **sintaxe** e o conjunto de circuitos de memória (objetos de memória) como **semântica**.
- Esta linguagem naturalmente compreendida pelo dispositivo eletrônico é chamada de **linguagem de máquina** ou **linguagem nativa** do computador.

- O conjunto de objetos de memória (semântica) da **linguagem de máquina** varia de acordo com a forma como os circuitos do computador estão organizados, isto é, da arquitetura interna de cada máquina.
- Por isso, cada modelo de computador tem sua própria linguagem de máquina.

- Assim, em princípio, para nos comunicarmos com computadores, teríamos de escrever nossas instruções por meio de linguagem de máquina.
- Isto é, interromper ou não a passagem de corrente pelos circuitos eletrônicos do computador de modo que ele, obedecendo às leis do eletromagnetismo, efetue o que desejamos.

Representação matemática das linguagens de máquina

- Representamos por 0 o estado fundamental em que o computador se encontra ligado, isto é, há passagem de corrente elétrica por um circuito.
- O estado em que não há passagem de corrente elétrica é denotado por 1.
- Assim, o estado de qualquer circuito de memória pode ser representado por uma sequência (ou cadeia) de 0's e 1's.

Sistema de numeração binário e linguagem de máquina

- Um sistema de numeração é um conjunto de regras para representar números.
- A regra mais fundamental de qualquer sistema de numeração é dada pelo conjunto de algarismos usados para representar números. Por exemplo, no sistema decimal, utilizam-se 10 algarismos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9).
- Assim, uma cadeia de algarismo formada pelos elementos acima é uma representação de um número na base decimal.

- Igualmente, portanto, uma cadeia de de 0's e 1's pode ser entendida como a representação de um número em um sistema de numeração que utiliza somente os algarismo 0 e 1.
- Tal sistema é bem conhecido e denominado **sistema binário**.
- Deste modo, qualquer estado da memória pode ser representado por um número binário ou, o que é o mesmo, por uma cadeia de 0's e 1's.

Instruções nativas

- Algumas operações que desejamos realizar são muito corriqueiras e básicas para a definição de operações mais complexas. Por exemplo, adição é base para a construção da multiplicação.
- Tais operações básicas são costumeiramente implantadas diretamente nos circuitos eletrônicos do computador para poupar aos seus usuários o trabalho de repetidamente escrever escrever instruções para que tais operações sejam realizadas.

- Estas operações pré-gravadas nos circuitos são chamadas de **instruções nativas** do computador.
- Atualmente, as linguagens de máquinas usam estas instruções nativas como seu léxico estendido.

Linguagens de computador

- As instruções nativas da linguagem de máquina podem ser organizadas logicamente e representadas por símbolos mnemônicos para facilitar o trabalho dos usuários do computador,
- Tais mnemônicos com a lógica que os organiza forma o que se chama de **linguagem de montagem ou assembly** de um computador específico.

- Assim, mais geralmente, chamamos de **linguagem de computador** à sua linguagem de máquina ou qualquer outra uma linguagem formal (inclusive, o assembly) que possa ser traduzida por correspondência biunívoca para a linguagem de máquina de um computador.
- As linguagens de computador servem para permitir que humanos se comuniquem com as máquinas a fim de dar a elas instruções para serem executadas.

Programa

Chamamos **programa** a uma sequência de instruções escrita em linguagem de computador.

Linguagem de alto e baixo nível

- Diz-se que uma linguagem é de baixo nível se seus comandos e sintaxe são específicas à arquitetura de um determinado dispositivos. Assim, além da própria linguagem de máquina, as linguagens de montagem são também linguagens de baixo nível.
- Chama-se **linguagem de alto nível** a uma linguagem que se caracteriza por não depender da arquitetura de qualquer dispositivo eletrônico específico.
- Como consequência, linguagens de alto nível costumam ser construídas para se aproximar da sintaxe de alguma linguagem humana.

Tradutores: a implementação de uma linguagem não nativa

- Linguagens não nativas sejam elas de montagem ou de alto nível não são compreendidas diretamente pelo computador, pois este só compreende sua linguagem nativa.
- Por isso, para entender uma linguagem de alto nível (ou mesmo assembly) um computador requer um tradutor desta linguagem para sua linguagem nativa.
- Um tradutor é um programa previamente armazenados no computador que efetua a tradução de uma linguagem de montagem ou de alto nível para linguagem de máquina.

- Diz-se que os tradutores são a implementação de uma linguagem de computador não nativa
- Os tradutores classificam-se em: montadores (assemblers), interpretadores, compiladores e máquinas virtuais.

Montadores, interpretadores e compiladores

- Montadores são programas que traduzem os mnemônicos de uma linguagem de montagem para uma linguagem de máquina de um determinado dispositivo.
- Interpretador é um programa que traduz individualmente os comandos de uma linguagem de alto nível para linguagem de máquina e o executa.
- Compilador é um programa que traduz um conjunto instruções escritas em uma linguagem de alto nível (código fonte) para um programa em linguagem de máquina (código objeto ou executável).

Máquinas virtuais

Uma máquina virtual é um programa que simula um computador cuja linguagem de máquina é denominada de "bytecode". Uma máquina virtual é composta por:

- um compilador de uma linguagem de alto nível para bytecode;
- um interpretador ou compilador de bytecode para linguagem de máquina, e
- outros programas usados para controle da execução do programa na máquina virtual e interação com o sistema operacional.

Máquinas virtuais servem para:

- tornar a execução de interpretadores mais rápidas, tendo em vista que o código interpretado já foi verificado previamente por um compilador, e
- tornar os programas compilados para bytecode passíveis de serem executados por qualquer dispositivo independentemente de sua linguagem de máquina.

Programação sequencial

Programa e programação sequencial

Programa

Chamamos **programa** a um algoritmo escrito em linguagem de computador.

Como o algoritmos são seqüências de instruções, a transliteração desta seqüência diretamente para uma linguagem de computador é chamada de **programação sequencial**.

Programação sequencial na arquitetura de von Neumann

- Em primeiro lugar, é preciso armazenar os programas na memória do computador.
- Em seguida, precisamos separar endereços de memória para armazenar os dados que serão entrados na memória do computador. Esta operação é chamada de **alocação de memória**.
- Na sequência, executa-se a escrita na memória alocada dos dados do problema - **entrada de dados**.

- Então, realiza-se com os dados entrados a sequência de operações prescritas no algoritmo de solução do problema - **processamento de dados**.
- Por fim, executa-se a gravação do resultado do processamento em um dispositivo de **saída de dados**.

Alocação de memória

- Os dados que serão introduzidos no computador ficam armazenados na memória por meio do registro de presença ou ausência de sinal eletromagnético em dada região do circuito.
- Chama-se de bit a menor unidade do circuito elétrico capaz de estar individualmente carregada ou descarregada de sinal elétrico. Representamos um bit carregado por 0 e um bit descarregado por 1.
- Por isso, para o computador, não haveria diferença entre, por exemplo, uma letra e um número, já que ele representa ambos por meio de bits carregados ou não, isto é, números formados pelos algarismos 0 e 1.

- Assim, quando do processamento de dados, o computador poderia, por exemplo, somar uma letra com um número sem identificar qualquer erro!.
- A fim de resolver esse problema foi criado o conceito de **tipo de dado** que apresentamos a seguir

Tipos de dados

Definição

Tipo de dado é a especificação do conjunto de elementos que podem ser armazenados em um dado na memória bem como as operações podem ser realizadas com estes elementos.

Cada linguagem de programação define os tipos de dados que ela é capaz de manipular.

Veja alguns exemplos de tipos de dados comuns a muitas distintas linguagens de programação:

- **booleano** (ou **lógico**) com as operações **conjunção**, **disjunção** e **negação**.
- **inteiro** ou **ponto fixo** com as operações **adição**, **multiplicação** e **exponenciação**.
- **real** ou **ponto flutuante** com as operações inteiras adicionadas de **divisão**.
- **complexo** com as operação reais adicionadas de **radiciação**.
- **character** com as operações de **concatenação** e **ordenação**.

- No momento da alocação de memória, informa-se ao computador o tipo de dado que será armazenado no endereço de memória especificado.
- Desta forma, no momento do processamento de dados, o processador saberá previamente que operações poderão ser feitas.
- Se o programador introduzir uma instrução que exija a execução de uma operação que não pode ser realizada com os tipos de dados em memória, o processador informará que há um erro e interromperá a execução do programa.

Entrada de dados

- Entrada (ou leitura) de dados é o procedimento de introduzir dados que representam algum fenômeno do mundo concreto dentro da memória de um computador.
- A entrada de dados é realizada por meio de um dispositivo de entrada de dados, copiando dados introduzidos por um usuário na memória (*buffer*) de um dispositivo de entrada para um endereço de memória do computador.
- Os dados introduzidos devem pertencer a algum tipo de dado conhecido pela linguagem de programação utilizada no momento da procedimento de entrada de dados.

Processamento de dados

- É uma sequência lógica e finita de operações realizadas sobre os dados lidos pelo programa, seguindo um algoritmo pré-determinado pelo programador.
- As operações realizadas sobre os dados lidos devem necessariamente respeitar às operações definidas para o tipo de dado armazenado em memória.
- Por exemplo, na maior parte das linguagens de programação de alto nível, não se pode realizar a operação de **concatenação** de dois **números inteiros**, pois esta operação normalmente não está definida para inteiros.

Saída de dados

- É o procedimento de copiar dados disponíveis em algum endereço de memória do computador para a memória (*buffer*) de um dispositivo de saída de dados.
- A ação de copiar da memória do computador para o buffer do dispositivo de saída de dados também é conhecida como escrita de dados no dispositivo de saída.

Código fonte, Código Objeto e Run Time

- O texto de um programa escrito em uma linguagem de alto nível é chamado de **código fonte** e o resultado de sua tradução para linguagem de máquina de **código objeto**.
- Se o código objeto não precisa passar por nenhuma outra transformação para ser executado, ele é um executável.

- **Runtime** ou **biblioteca de runtime** (às vezes, chamada em Português de tempo de execução) é um conjunto de instruções escritas pelo desenvolvedor de uma linguagem de programação de alto nível que visam a permitir a comunicação entre o código objeto e o sistema operacional do computador a fim de tornar o código objeto em executável.

Referências

Referências



FOROUZAN, B. e MOSHARRAF, F *Fundamentos da Ciência da Computação*. Tradução da 2a. ed. internacional, São Paulo: CENGAGE Learning, 2011.



BLUME, L.; EASLEY, D.; KLEINBERG, J.; KLEINBERG, R.; TÁRDOS, E., *Introduction to computer science and economic theory*, Journal of Economic Theory, vol. 156, 2015, pp. 1-13, Acesso em 09/11/2018, Disponível em [link](#).