

Introdução

Área de Conhecimento em Algoritmos e Teoria - DCC/UFMG

Fundamentos de Teoria da Computação

2021/2

O que é a “Teoria da Computação”?

Introdução - O que é “computação”?

- Vamos abrir o curso de Teoria da Computação com uma pergunta:

Como podemos definir o conceito de “computação”?

Algumas tentativas de resposta são:

- É o ato de raciocinar seguindo passos lógicos e matemáticos?
- É o ato seguir um algoritmo, recebendo entradas e produzindo saídas?
- É exatamente aquilo que um computador faz?
- ...

Introdução - O que é “computação”?

- E se definirmos “computar” como “aquilo que um computador faz”, então temos uma outra pergunta para responder:

O que é um “computador”?

Em particular, podemos nos perguntar:

- Uma calculadora de bolso é um computador?
- Um smartphone é um computador?
- Nosso cérebro é um computador?
- O universo é um computador?
- O que não seria um computador?

Introdução - O que é “computação”?

- Vamos abordar a questão sobre o que é “computação” começando por uma observação menos controversa.

Intuitivamente, todos podemos concordar que “**computar**” está relacionado a responder perguntas como:

- Pergunta: Quanto é $6 + 5$?
 - Resposta: 11.
- Responder a uma pergunta como a acima parece ser o tipo de coisa que todos concordamos em chamar de “computar”.

Introdução - O que é “computação”?

- Mas então, vamos considerar uma pergunta bem semelhante à primeira, mas um pouco “mais complicada”:
 - Pergunta: Quanto é cento e vinte e dois milhões, trezentos e vinte e seis mil, setecentos e oitenta e nove mais quatrocentos e cinquenta e um milhões, setecentos e setenta e sete mil, trezentos e vinte e dois?
 - Resposta: Quinhentos e setenta e quatro milhões, centro e quatro mil, cento e onze.
- Responder a pergunta acima “de cabeça” pode ser possível para um gênio da aritmética.

Logo, parece natural concluir que:

“O cérebro de um gênio da aritmética consegue computar.”

Introdução - O que é “computação”?

- Mas quem não é um gênio da aritmética pode computar também?

Sim, basta:

1. converter o problema para uma representação adequada:
 $122\ 326\ 789 + 451\ 777\ 322$;
2. escrever a representação no papel;
3. efetuar a soma dígito a dígito, obtendo a resposta: $574\ 104\ 111$;
4. converter a representação para sons, e dizer em voz alta.

- Oba, então eu também posso computar!

(Mesmo que precise de ajuda de lápis e papel, e um pouco de tempo...)

Logo, também parece natural concluir que:

“O cérebro de uma pessoa normal, com uma ajudinha de lápis e papel, também consegue computar.”

Introdução - O que é “computação”?

- Então, até agora decidimos que “computar” é algo que:
 - pode ser feito diretamente dentro de um cérebro humano (se você for um gênio da aritmética), ou
 - pode ser feito por um cérebro humano munido de recursos externos, como lápis e papel (se você for uma pessoa normal, como o Mário).
- Mas será que “computação” é algo que pode ser feito apenas por cérebros (humanos)?

Ou será que também seriam capazes de “computar”:

- (o cérebro de) um macaco corretamente adestrado?
- uma máquina apropriadamente projetada e construída?
- fótons cuidadosamente excitados segundo a mecânica quântica?
- moléculas orgânicas (como DNA ou RNA) reagindo em substrato adequado?
- ...

Introdução - Modelos de computação

- Se você respondeu “sim” a mais de uma pergunta acima...
 - você considera que o ato de “computar” não necessariamente está limitado a instâncias particulares de “quem” ou “o quê” está “computando”;
 - ou seja, você acha que “computar” pode transcender o conceito de um “computador” específico.
- Qual seria, então, a propriedade essencial em comum entre os todos os atos que você considerou como “computação”?

Em outras palavras:

“Qual é o modelo de computação que captura a essência do ato de “computar” presente em todos os atos acima?”

- Um **modelo de computação** é uma abstração do ato de “computar” que captura a essência compartilhada por todos os atos considerados como “computação”.

Naturalmente, um bom modelo deve deixar de fora tudo que não é considerado “computação”.

- **Modelo funcional de computação:**

“Computar é calcular uma função que mapeia entradas a saídas.”

Neste modelo, se dois procedimentos calculam a mesma função, então eles são o mesmo procedimento (ou algoritmo).

- **Modelo imperativo de computação:**

“Computar é seguir uma sequência de comandos bem delimitada, manipulando representações.”

Neste modelo, “entradas” e “saídas” não são necessariamente tão claras.

- Consideramos que um vídeo game e um SO “computam”, mas quem é a “entrada” e a “saída” de cada um deles?
- Existem ainda outros modelos de computação: **lógico**, **probabilístico**, etc.

Introdução - Modelos de computação

- A verdade é que não existe (ainda?) uma resposta matemática definitiva sobre o que é “**computação**”.

(Apesar de, como veremos, existirem conjecturas bem sólidas...)

Assim como não existe uma resposta matemática definitiva (ainda?) sobre:

- o que é “**inteligência**”?
- o que é “**consciência**”?
- Então, a seguinte pergunta:
 - Pergunta: É possível criar máquinas que “computem”?não está bem formulada, e deve ser substituída por:
 - Pergunta: É possível criar máquinas que implementem um dado modelo de computação?
 - Spoiler: Sim! Este é o tema deste curso!
- Já as questões de se que podemos criar máquinas que apresentem inteligência e/ou consciência ficam pra depois...

Teoria da Computação: Definição

- A **teoria da computação** é o subcampo da ciência da computação e da matemática que busca determinar:
 - quais problemas podem ser computados,
 - utilizando quais recursos,
 - em um dado modelo de computação.
- Por milhares de anos, a computação foi feita com lápis e papel, ou giz e quadro, ou mentalmente, às vezes com a ajuda de tabelas.

Computadores são uma invenção recente!

- Normalmente um modelo de computação é desenvolvido antes de uma máquina que o implemente ser construída.

A **Teoria da Computação Quântica**, por exemplo, foi bastante desenvolvida antes mesmo que os computadores quânticos fossem construídos.

(E o poder previsto teoricamente para estas máquinas é uma das razões pelas quais tanto dinheiro tem sido investido em sua construção.)

Teoria da Computação: Importância

- A Teoria da Computação é crucial para podermos entender a dificuldade intrínseca de problemas.
- Por enquanto, vamos definir um problema como:
 - “**fácil**”, se uma máquina “simples” é capaz de resolvê-lo.
 - “**difícil**”, se é necessário uma máquina “complicada” para resolvê-lo.

Por exemplo, intuitivamente, um ábaco é “mais simples” que uma calculadora de bolso, que é “mais simples” que um *laptop* de última geração.

- Determinar a dificuldade intrínseca de cada problema é relevante porque selecionar a máquina mais simples que dê conta de resolver um problema pode economizar recursos.

Por exemplo, o chip do seu relógio de pulso não tem o mesmo poder que a CPU do seu *laptop* de última geração...

... e seria um desperdício de recursos se tivesse!

Teoria da Computação: Importância

- Como você ordenaria os problemas abaixo do
 - “mais fácil” (i.e., que necessita da máquina “mais simples”) para o
 - “mais difícil” (i.e., que necessita da máquina “mais complexa”)?
- (A) Dado um número natural, decidir se ele é primo.
- (B) Dada uma palavra em português, decidir se sua grafia está correta.
- (C) Dado um programa em linguagem C, decidir se ele não tem erro de sintaxe.
- (D) Dada uma fórmula em lógica de predicados, decidir se ela é verdadeira ou falsa.
(Ex: a fórmula $\forall x \exists y : P(x) \rightarrow Q(y)$.)

Como vamos ver neste curso, a “ordem de dificuldade” dos problemas é:

- (B) é o “mais simples”: um **autômato finito** resolve o problema.
- (C) é o “segundo mais simples”: um **autômato com pilha** resolve o problema.
- (A) é o “segundo mais difícil”: uma **máquina de Turing** resolve o problema.
- (D) é o “mais difícil”: não existe máquina que resolva este problema no caso geral!

O que é este curso?

Objetivos do curso

- Neste curso vamos explorar alguns **modelos teóricos de computação**, que são a base de todos os computadores existentes, respondendo às seguintes perguntas:
 - “Computadores existem?”
É possível construir máquinas que resolvam problemas automaticamente?
 - “Para quais tipos de problemas existem computadores?”
Que tipos de máquinas somos capazes de construir, e quais tipos de problemas elas são capazes de resolver?
 - “Como sabemos se já inventamos o melhor computador possível?”
Há limites para os tipos de problemas resolúveis por nossas máquinas?

Objetivos do curso

- Tópicos abordados neste curso:
 - **Teoria de autômatos:** como criar modelos matemáticos de computação.
 - **Linguagens formais e gramáticas:** como descrever e especificar problemas computacionais.
 - **Decidibilidade:** como determinar se um problema pode ser resolvido em um modelo de computação.
- Tópicos não abordados neste curso:
 - **Complexidade computacional:** como determinar o quanto de recursos (tempo/espaço) é preciso para resolver um problema em um dado modelo computacional.

- Algumas áreas de aplicação da Teoria da Computação:
 - implementação de compiladores;
 - verificação formal de código;
 - inteligência artificial;
 - processamento de linguagem natural;
 - reconhecimento de padrões;
 - engenharia de software (UML);
 - computação genômica (sequenciamento de DNA);
 - segurança em redes (protocolos);
 - teoria de jogos;
 - etc.