

Teoria da Computação

Aula Teórica 1: Motivação

António Ravara e Vasco Amaral

Departamento de Informática

5 de Março de 2018

Teoria da Computação?!

Depois das diversas cadeiras de programação leccionadas no curso até agora podemos ficar com a sensação que:

- ▶ se poderá escrever programas para qualquer problema computacional
- ▶ mesmo sendo trabalho árduo, difícil de aprender, podendo implicar muito debug, com tempo e esforço podemos escrever qualquer programa

Surpreendentemente isto não é verdade!

Existem problemas computacionais para os quais não se consegue fazer um programa.

Teoria da Computação?!

O que vamos ver hoje:

- ▶ De que trata a disciplina?
- ▶ Porque é relevante para a Informática?

O que é a Teoria da computação?

Sinteticamente: estudo matemático da computação.

Por outras palavras, a “matemática” (i.e., a base rigorosa e formal) da informática!

Relevância da computação

Clarifica tanto o poder como os limites do que faz um computador...

Teoria da Computação?!

Para que queremos *teorias*?

- ▶ Ao analisar um problema, devemos concentrar-nos no essencial
- ▶ Criamos um *modelo* - fundamentos sólidos, pela abstracção
A Biologia, a Química, a Física, a Matemática procedem assim
(para explicar a matéria ou a vida)
“Teorias” poderosas: a relatividade ou a evolução são exemplos
- ▶ A Computação cria modelos para entender a informática
O que faz um computador?
O que é um programa? Tudo é programável?

Na verdade, antes de haver informática houve computação...
A computação surge na/da Matemática e cria a informática como
“derivado” (subproduto).

Matemática da/na Informática

A informática é formal?

- ▶ Baseia-se na lógica
No coração das máquinas estão processadores, que contêm circuitos, que executam operações lógicas...
- ▶ Instrumento central são línguas artificiais
 - ▶ As linguagens de programação têm regras de sintaxe e semântica rígidas
 - ▶ As máquinas (computadores) executam os programas rigida (ou cega) mente

Não há lugar para a informalidade...

Programa?!

Para que serve...

- ▶ Uma linguagem de programação?
 - ▶ Para escrever *programas* (textos de uma língua artificial)
 - ▶ Uma linguagem é o veículo para muitos programas, com objectivos diferentes
 - ▶ Muitas linguagens são *universais*: servem para escrever qualquer programa, para fazer *tudo o que se pode fazer...*
- ▶ Um programa?
Para implementar uma forma detalhada e precisa de instruir uma máquina para executar um *algoritmo*.

Algoritmo?!

Ideia intuitiva

Procedimento genérico e reproduzível para fazer determinado cálculo (em tempo finito).

Exemplos

- ▶ Os mais antigos que se conhecem foram desenvolvidos na Babilónia, para calcular a área de triângulos.
- ▶ Na Grécia antiga desenvolveram-se vários:
 - o de Euclides para o MDC;
 - o crivo de Eratóstenes para listar primos até dado valor, ...

Algoritmo?!

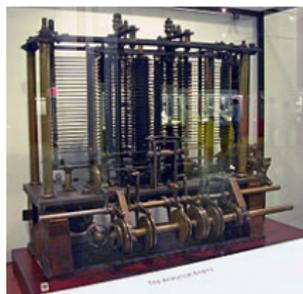


©Public Domain - Wikimedia Commons

Al-Khowarizmi

Nome de matemático Persa (actual Iraque) do séc IX,
Escreveu um tratado de álgebra, com métodos de cálculo, alguns da China e da Índia. A propósito, cálculo quer dizer pedra – usavam-se nos ábacos...

Automação



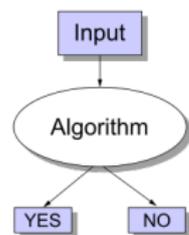
©Public Domain - Wikimedia Commons

Automação

Surgiu a partir do séc XVII a vontade de construir máquinas para executar algoritmos (concretos).

- ▶ Surgem os relógios mecânicos e os teares
- ▶ Pascal e depois Leibniz constroem calculadoras aritméticas mecânicas
- ▶ Babbage cria uma máquina analítica programável

É uma máquina universal? A essência da computação



©Public Domain - taken from Wikipedia

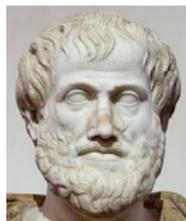
O problema da decisão

É possível determinar se um dado elemento de um universo satisfaz uma dada propriedade?

Algoritmo

- ▶ O procedimento efectivo para resolver um problema de decisão
- ▶ A tentativa “geral” de resolver o problema na Lógica Matemática foi um falhanço espectacular!
- ▶ Igualmente um sucesso: a máquina universal (o computador)!

Tudo começa com a lógica...



©Public Domain - Taken from wikipedia

Aristóteles: a lógica dedutiva

- ▶ Viveu em Atenas de 384 a 322 A.C.
- ▶ Definiu as bases axiomáticas do raciocínio (lógico).
- ▶ Junto com Sócrates e Platão, estabeleceu as bases do conhecimento científico da Europa.

Tudo começa com a lógica...

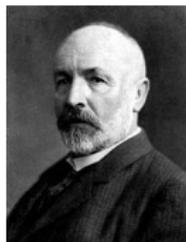


©Public Domain - Taken from Wikipedia

Euclides: a prova matemática

- ▶ Viveu em Alexandria, cerca do ano 300 A.C.
- ▶ Formalizou a Geometria, baseada em princípios matemáticos rigorosos.
- ▶ O seu livro "Os elementos" foi referência até ao Séc XIX.

A formalização da Matemática - Avanços no Séc. XIX



©Public Domain - taken from Wikipedia

Georg Cantor: a teoria de conjuntos

- ▶ Nasceu na Rússia em 1845; morreu na Alemanha em 1918.
- ▶ Inventou a teoria de conjuntos, definiu a noção de infinito e de números transfinitos; mostrou que há mais reais que naturais.

David Hilbert (apresentado mais à frente) disse em 1926:
“Ninguém nos expulsará do paraíso que Cantor criou para nós”

A formalização da Matemática



©Public Domain - taken from Wikipedia

Gottlob Frege: os quantificadores

- ▶ Viveu na Alemanha de 1848 a 1925.
- ▶ Inventou as noções de quantificadores e a Lógica de Segunda Ordem, permitindo o tratamento rigoroso das noções de funções e variáveis, e tendo formalizado a Aritmética.

Problemas no paraíso?!



©Public Domain - taken from Wikipedia

Bertrand Russell: a Lógica como fundação da Matemática

- ▶ Viveu em Inglaterra de 1872 a 1970.
- ▶ Escreveu "The Principles of Mathematics" mostrando que a Matemática pode ser deduzida de um pequeno número de princípios.

Problemas no paraíso?!



©Public Domain - taken from Wikipedia

Bertrand Russell: a Lógica como fundação da Matemática

- ▶ Estudou os fundamentos da Matemática e descobriu o paradoxo que abalou a teoria de conjuntos.
"O conjunto de todos os conjuntos contém-se a si mesmo?"
Se não, não tem todos os conjuntos;
se sim, a "definição" é circular...

A metamatemática - pode ou não usar-se a teoria de conjuntos?



©Public Domain - taken from Wikipedia

David Hilbert: a “prova automática”

- ▶ Viveu na Alemanha de 1862 a 1943.
- ▶ Definiu os Espaços de Hilbert, a matemática da relatividade e da quântica (“A Física é demasiado difícil para os Físicos”), a Teoria da Prova e a Lógica Matemática.
- ▶ Em 1900 estabeleceu os 23 mais importantes problemas da Matemática, alguns deles estão ainda hoje por resolver.

A metamatemática - pode ou não usar-se a teoria de conjuntos?



©Public Domain - taken from Wikipedia

David Hilbert: a “prova automática”

- ▶ Definiu um programa de investigação para formular TODA a Matemática com bases sólidas e assentes na Lógica (a partir de um conjunto finito de axiomas, provados consistentes).
- ▶ Afirmou mesmo: “Temos que saber, viremos a saber!” para sublinhar que a Matemática desvendaria toda a Verdade.

O “Programa de Hilbert”

O desafio de Hilbert

Fundamentar a Matemática:

- ▶ Tudo devia ser formulado e deduzido a partir de bases sólidas e assentes na Lógica.
- ▶ Para tal, era necessário encontrar um conjunto finito de axiomas:
 - ▶ consistentes – não permitindo concluir contradições
 - ▶ completos – toda a afirmação ou é verdadeira ou é falsa
 - ▶ decidíveis – existem métodos para verificar a veracidade das afirmações

A consciência dos limites



©Public Domain - taken from Wikipedia

Kurt Gödel: a consistência e a (in)completude

- ▶ Nasceu na Austria em 1906; morreu nos EUA em 1978.
- ▶ Mostrou a completude da Lógica de primeira ordem (no doutoramento, aos 24 anos) e a incompletude de qualquer sistema axiomático capaz de descrever a aritmética dos naturais.

Kurt Gödel: a consistência e a (in)completude

Teoremas

1. Incompletude: num sistema formal axiomático *consistente*, com (pelo menos) o poder da aritmética, haverá sempre afirmações verdadeiras que não é possível provar. Em particular, não pode provar a sua própria consistência.
2. Inconsistência: um tal sistema, se é completo, não é consistente

Fim e início

- ▶ Anunciou os resultados no congresso de homenagem a Hilbert
John von Neumann diz "está tudo acabado!"
- ▶ Estabeleceu as bases da Computação – Teoria das funções recursivas – definindo o que é decidível.

Do fracasso do programa de Hilbert...

- ▶ Incompletude: há proposições verdadeiras num sistema consistente que não podem ser provadas.
- ▶ Indecidibilidade: há proposições que não se conseguem determinar a sua veracidade

O paradoxo do mentiroso – Epimenedes

Só digo mentiras!

- ▶ é verdade? então menti (não digo só mentiras)...
- ▶ é falsa? então disse a verdade?!

O nascimento da computação



©Public Domain - taken from Wikipedia

Alonzo Church: as linguagens de programação

- ▶ Viveu nos EUA de 1903 a 1995.
- ▶ Contribuiu para a Lógica Matemática e para a Teoria da Computação, tendo definido o Cálculo-Lambda para provar que é indecidível determinar se dada proposição arbitrária de uma teoria matemática é ou não verdadeira.
- ▶ A simplicidade e expressividade do Cálculo-Lambda, que permite codificar funções, fizeram dele a primeira linguagem de programação.

"The imitation game"



©Public Domain - taken from Wikipedia

Alan Turing: heroi maldito...

- ▶ Viveu em Inglaterra de 1912 a 1952.
- ▶ Descodificou o sistema de encriptação da marinha Alemã. Churchill disse que não é um homem que ganha uma guerra, mas que sem Turing a Grã-Bretanha teria provavelmente perdido a segunda guerra mundial.

Mas "the Bombe", a máquina que descodificava as mensagens da Enigma, não surgiu por acaso...

O que se pode (e não pode) fazer com um computador?

Alan Turing: a computação

- ▶ *O que se pode fazer*: um modelo de computação
Uma "máquina" para executar qualquer programa (formalizando um algoritmo)!
a Máquina Universal de Turing.
- ▶ Estabeleceu a Tese de Church-Turing:
Toda a função algorítmicamente calculável é computável por uma Máquina de Turing.
- ▶ *O que não se pode fazer*
indecidibilidade do "Halting Problem":
Dado um programa arbitrário e o seu "input" (finito), pode-se decidir se o programa terminará?

Está criada a computação, há menos de 100 anos...

O computador: dos modelos aos protótipos.

Surgem as máquinas (e a IA)

Há 70 anos...

- ▶ Turing concebeu e construiu o primeiro computador europeu:
A partir de 1946, em Cambridge: o ACE
- ▶ Um pouco antes (1945), nos EUA (em Princeton)...
John von Neumann concebeu a arquitectura interna de uma máquina universal
e construiu o EDVAC, o primeiro computador digital.

Do que trata a cadeira?

Modelação com Conjuntos e Lógica

- ▶ Revisão dos Princípios de Indução Natural e Estrutural
- ▶ Revisão de Lógica de Primeira Ordem
- ▶ Revisão das noções de Conjuntos, Funções e Relações
- ▶ Finito e Infinito; argumento diagonal de Cantor.
- ▶ Diferença entre função e algoritmo.
- ▶ Modelos de sistemas simples e tipos abstractos de dados.

Do que trata a cadeira?

Autómatos e Especificações

- ▶ O que é um modelo computacional?
- ▶ Autómatos finitos deterministas e expressões regulares.
- ▶ Determinismo e não determinismo.
- ▶ Linguagens independentes de contexto e máquinas de pilha.
- ▶ Análise sintática (LL e LR).

Do que trata a cadeira?

Computabilidade

- ▶ Computabilidade.
- ▶ Expressividade computacional. Equivalência entre programação funcional e imperativa.
- ▶ Máquinas abstractas e níveis de interpretação.
- ▶ Universalidade Turing. Tese de Church-Turing. Indecidibilidade (da terminação).