UNIVERSIDADE
BEIRA INTERIOR

Teoria da Computação
Aula 0 - O nascer de uma disciplina

Simão Melo de Sousa

Prolegómeno

Para quê estudar o conceito de algoritmo?

para perceber e estudar a **informática**, o seu **poderio**, a sua **capacidade**

é preciso saber com precisão o que **são algoritmos**

mas também o que **não são**

o que **fazem**

mas também o que **não conseguem fazer**

Para quê estudar o conceito de algoritmo?

daí advém um **melhor domínio** da disciplina

uma melhor compreensão das **causas e consequências** do uso da informática

mas também a percepção das **revoluções tecnológicas** por acontecer

> *É preciso perceber o mundo [tecnológico] e não simplesmente submeter-se a ele, ou seja, apropriar-se dos fundamentos da informática, e não só o seu uso. (anónimo)*

... or shooting yourself in the foot

o conceito de algoritmo evoluiu com a **própria história da matemática**

a própria **informática** nasceu como **disciplina** quando a matemática, a lógica mais precisamente, conseguiu definir com **precisão o que era um algoritmo**...

... em resposta à crise dos fundamentos da matemática do início do século XX

deste momento fundador temos uma definição precisa do que é ou não é um algoritmo e da sua **associação** ao conceito de **máquina**

... mais detalhe mais adiante.

existem **limites à capacidade de resolução de problemas** por um computador, mesmo na hipótese *idealista* de ausência de restrição em termos de tempo (de execução) e de espaço (memória)

Para **delinear esse limites**, visaremos nesta UC:

- **perceber a capacidade de computação das máquinas**, assim como os seus limites teóricos. Precisaremos de definir formalmente o que é e o que não é um algoritmo, um programa;
- **perceber os conceitos que fundamentam as linguagens de programação**. Precisaremos de determinar e estudar formalmente as construções que determinam a expressividade das linguagens de programação assim como o comportamento dos programas.

Introdução

uma primeira tentativa de definição

(o que se diz por este mundo fora)

um conjunto de **encantações mágicas** com vida própria, temidas por muitos, proferidas por poucos — **obscuros feiticeiros**, para **resolver ou criar *ubi*quamente todos os problemas** do mundo

nada disso...

vejamos como o vocabulário e o conceito evoluiu a par da história da matemática

para testar a primalidade podemos optar pela estratégia seguinte:
Para um determinado número inteiro n maior do que 1

- tentar a divisão inteira de n por 2
  $\Longrightarrow$ Se o resto for 0 então n é divisível por 2, logo n não é primo
- Senão, experimentar a divisão por 3
  $\Longrightarrow$ Se o resto for 0 então n é divisível por 3, logo n não é primo
- etc... repetir o processo até $n-1$
  $\Longrightarrow$ Se o resto for 0 então n é divisível, logo n não é primo
- Se nenhuma divisão resultou então n é primo

este processo é muito elementar (existem métodos bem mais interessantes)

embora simples, este exemplo é esclarecedor:

1. esclarece bem a **estrutura** do tratamento efectivo
2. esclarece bem a **natureza** do tratamento efectivo

um algoritmo é formado por

- entradas e saídas
- iteração
- composição
- decisão,
- memória/estado interno
- acções atómicas
- reutilização/modularidade

- tentar a divisão inteira de n por 2
  $\Longrightarrow$ Se o resto for 0 então n é divisível por 2, logo n não é primo
- Senão, experimentar a divisão por 3
  $\Longrightarrow$ Se o resto for 0 então n é divisível por 3, logo n não é primo
- etc... repetir o processo até $n - 1$
  $\Longrightarrow$ Se o resto for 0 então n é divisível, logo n não é primo
- Se nenhuma divisão resultou então n é primo

- apresenta-se como um conjunto **finito** de instruções
- funciona para a **classe** de todos os n possíveis

  (o caso $n = 5$ é uma **instância**)
- execução **finita**
- execução absolutamente **sem ambiguidades**, **determinista**, passos básicos claros, simples e inequívocos
- **correcta**

- tentar a divisão inteira de n por 2
  $\Longrightarrow$ Se o resto for 0 então n é divisível por 2, logo n não é primo
- Senão, experimentar a divisão por 3
  $\Longrightarrow$ Se o resto for 0 então n é divisível por 3, logo n não é primo
- etc... repetir o processo até $n - 1$
  $\Longrightarrow$ Se o resto for 0 então n é divisível, logo n não é primo
- Se nenhuma divisão resultou então n é primo

> *Um algoritmo é, simplesmente, uma forma de descrever* ***em todos os seus detalhes como proceder*** *para fazer alguma coisa. Acontece que muitas acções mecânicas, provavelmente todas, se prestam bem a tal escrutínio. O objectivo é* ***esvaziar o cálculo de todo o pensamento****, com a finalidade de* ***o tornar executável por um mecanismo*** *numérico (computador...). Trabalhamos assim unicamente com um reflexo numérico do sistema real com o qual o algoritmo interage*

Gérard Berry (1948-)

... pressupõe o uso de um computador *moderno*

Aos que já descrevemos, Donald Knuth juntou o seguinte requisito

> ***efetividade*** *: todas as operações que o algoritmo deve realizar devem ser suficientemente básicas para poder ser executadas em princípio num intervalo de tempo finito por um ser humano com recurso a um papel e a uma caneta*

Donald E. Knuth, (1938-)

... pressupõe (a procura sem falha de) uma eficiência exemplar

nota sobre a estrutura de um algoritmo

algoritmo =	
sequência de instruções	quem a define? como está definido? como está arquivado e consultado?
+ **os dados de entrada**	
+ **os dados internos** do programa	a que nível pertence? como evoluem? quem os arquiva/manipula?
+ **processo de execução**	quem? o quê? como?

ao longo da história (e da procura da definição *última*), a **definição/relevância** destas componentes se **alterou**, como o veremos

pense, por ex. num **computador humano** (auxiliado ou não com uma **caneta e folha de papel**, com uma **calculadora básica**...) ou um **computador** na sua definição moderna

conceito de algoritmo vai adaptando-se conforme os **aparelhos disponíveis** que **podem executar** cálculos

não confundir **programa** e **algoritmo**

como o veremos, mais do que 5000 anos de algoritmia sem sequer um computador!

sem sequer uma definição exacta do termo algoritmo!

programa = concretização de um algoritmo por forma a que este possa ser entendido e executado por um mecanismo (e.g. computador, na sua definição moderna)

programar = implementar algoritmos na forma de programas

desenhar programas eficientes... **toda uma arte em si!**

Da história ao nascimento registado

é uma derivação do nome do matemático persa

Muḥammad ibn Mūsā al-Khwārizmī: 780 (actual Uzbequistão) - c.850 (actual Bagdad)

para celebrar a sua obra e popularização dos processos de cálculo sistemático na aritmética que apresentou na sua obra

a sua obra ambicionava resumir *enciclopedicamente* o conhecimento matemática indo-árabe, (re)organizá-la e projectá-la

objectivo atingido

devemo-lhe...

... o nascimento da álgebra moderna, a incógnita x, a popularização do sistema numerico decimal posicional, as bases da aritmética moderna, etc...

algoritmo: etimologia e origem

o termo **algorismus** aparece no século XII, como resultado da tradução ou vulgarização da obra dele

***Algorithmi**de numero indorum* (tradução para o Latim da obra sobre o sistema decimal posicional, onde abusadamente se traduz *processo de cálculo* por uma *latinização* do nome do autor)

*Carmen de **Algorismo*** (Alexandre de Villedieu, 1240)

> *Haec algorismus ars praesens dicitur, in qua*
> *Talibus Indorum fruimur bis quinque figuris.*
>
> *. . .**Algorismo** é a técnica na qual, presentemente, usamos as seguintes figuras indianas que representam dois vezes cinco.*

(um poema que introduz uma lição de aritmética algorítmica)

papel central de Leonardo de Piza (1170-1250, conhecido sob o nome de Fibonacci e considerado o maior matemático ocidental da sua geração) na divulgação do trabalho de Muḥammad ibn Mūsā al-Khwārizmī

passo sob silêncio as contribuições maiores do Fibonacci (e.g. a sequência e a sua história, etc.)

recordo aqui só

- o **conflito entre Abacistas e Algoristas** em que ele teve um papel fundamental (e polémico - a Igreja apoiava o sistema numérico romano)

  Leonardo de Pisa (*Liber Abaci*, i.e. **libertar-se do ábaco**) acreditava que o cálculo (no sistema numérico decimal posicional) era iminentemente mais poderoso e promissor do que a sistematização possível com os ábacos

  (... na relação entre ciência/matemática com engenharia "*da altura*", Leonardo advogava que a prática corrente desta última limitava a primeira)

- ... e o impacto prático imediato do seu trabalho: contabilidade, finanças, calculo de taxas e de impostos

  este aspecto pragmático (a automação como meio para melhorar o dia a dia) é constante na história do conceito

Da história ao nascimento registado

---

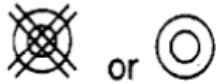
escribas, contabilistas como computadores

Suméria, 5000... 3400 - 2000 AC

das primeiras civilizações a desenvolver a escrita
(a par da Egípcia antiga)

os testemunhos chegam-nos na forma de barras de argila

desenvolveram um sistema numérico aditivo em base 10
e 60

1	10	60	600	3600	36000
					 or

163:

 60 60

 10 10 10 10 1 1 1

os sistemas numéricos em uso são o reflexo das necessidades sociais

nesta altura: sociedade agrária, gestão da áreas de cultivo e das colheitas, partilha de bens, contar o gado, calcular o tempo (as estações, o tempo das colheitas, etc...), resolver contendas comerciais/territoriais, etc

e.g. em caso de conflito, os agentes da lei decidam "objectivamente" com os algoritmos/cábulas

a base 60 é muito cómoda: 60 tem muitos divisores

$$2,\ 3,\ 4,\ 5,\ 6,\ 10,\ 12,\ 15,\ 20\ e\ 30$$

processos que precisam de multiplicações ou de divisões *usuais* são geralmente fáceis

mas... e o 7?

algoritmo para a partilha da colheita

como partilhar o resultado da colheita contada em celeiros e tendo em conta a justa medida (7 silà) que cada trabalhador merece? Quantos trabalhadores serão contemplados?

Tablet 50

1 granary of grain	36000 36000 36000 36000
silà 7	3600 3600 3600 3600 3600
each man receives	600 600 60 600 600 60
these men are	10 10 10 10 10 1
	remainder 3 silà of grain

celeiro = 2400 gur
gur = 480 silà
celeiro = 1 152 000 silà
silà $\approx$ um litro

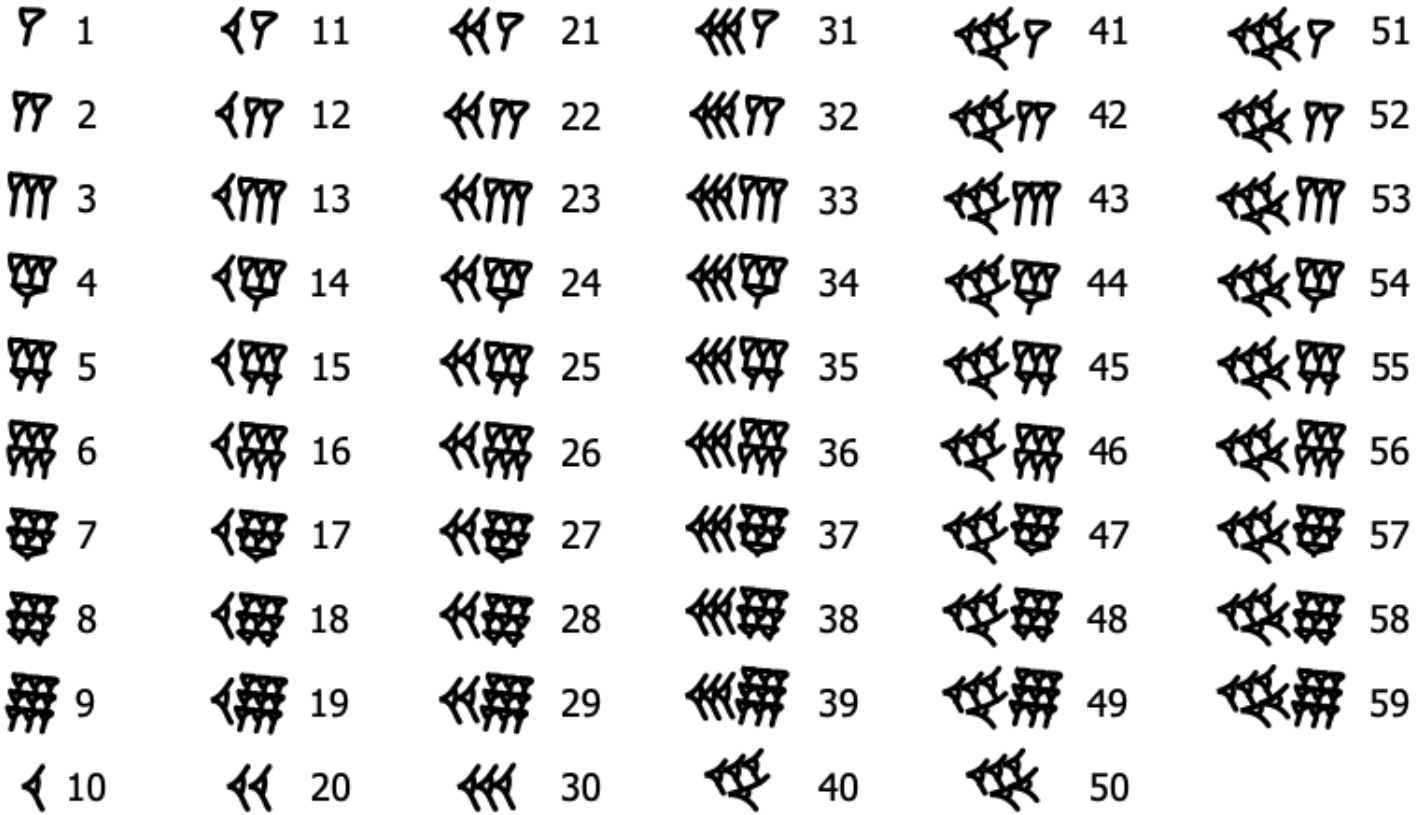
164571 homens (resto, 3 silà)

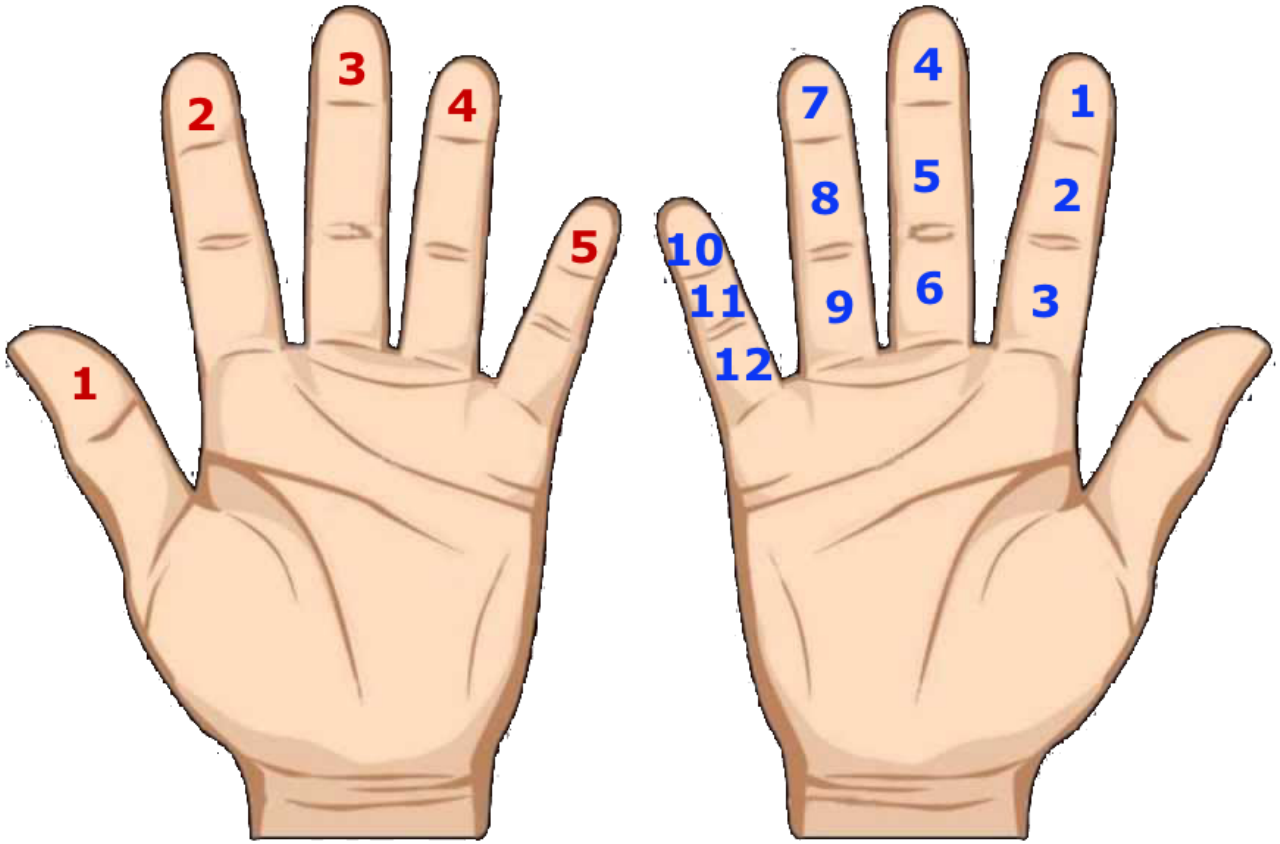
estas barras de argila (meros recibos) sugerem a existência de um algoritmo para a divisão (executado pelo contabilista)

... e na Babilónia antiga (3000... 2500-500... 170 AC)

a Babilónia antiga reaproveitou e estendeu parte do sistema numérico sumério

vigorava assim um sistema posicional de base 60 com uma notação explícita para a unidade e a dezena





12 e 60 de forma cómoda

astronomia babilónica: a observação do tempo centrado nos ciclos lunares e solares

o mês = um ciclo lunar $\approx$ **30 dias**

um ano = um ciclo solar

ora, a observação solar *aproxima* o ciclo solar a 12 ciclos lunares, **12 meses num ano**, **360 dias**

cada dia um grau, num ano o **círculo** todo é feito, em **360 graus**

hora : comodamente cada dia (e cada noite) é dividido em 12 unidades (o que se conta com uma só mão, que comodamente se divide por 2, 3, 4 e 6)

minutos e segundos = cada hora é dividida em 60 unidades por si também são divididas em 60 sub-unidades

um pouco mais sobre a numeração na Babilónia

a matemática na Babilónia não considerou inicialmente o 0 mas considerava os **reais**!

o 0 era inicialmente implicitamente denotado por um espaço

outro factor interessante é a ausência da informação das potências de 60 usadas na representação: é um sistema numérico posicional **com twist**
a sequência **2 13 20** pode tanto representar o valor $2 \times 60^2 + 13 \times 60 + 20$ como o valor $2 \times 60^5 + 13 + \frac{20}{60^2}$, ou seja genericamente os valores
$2 \times 60^m + 13 \times 60^n + 20 \times 60^p$ com $m > n > p$ relativos

é implicitamente assumido que a **desambiguação** é feita por **quem executa/lê**
i.e. **conhece** os valores m, n, p
$\Longrightarrow$ notação com vírgula flutuante (sem representação do expoente) bem antes dos **floats**

... estranhamente, usaram a bom proveito estes factores para desenvolver mecanismos de cálculos complexos

para calcular $\frac{a}{b}$ preferir multiplicar: $a \times \frac{1}{b}$

desenvolveram métodos para calcular o inverso $\frac{1}{b}$ para cada valor b pertinente

2	-	30	16	-	3:45	45	-	1:20
3	-	20	18	-	3:20	48	-	1:15
4	-	15	20	-	3	50	-	1:12
5	-	12	24	-	2:30	54	-	1: 6:40
6	-	10	25	-	2:24	1	-	1
8	-	7:30	27	-	2:13:20	1: 4	-	56:15
9	-	6:40	30	-	2	1:12	-	50
10	-	6	32	-	1:52:30	1:15	-	48
12	-	5	36	-	1:40	1:20	-	45
15	-	4	40	-	1:30	1:21	-	44:26:40

.... e 7?

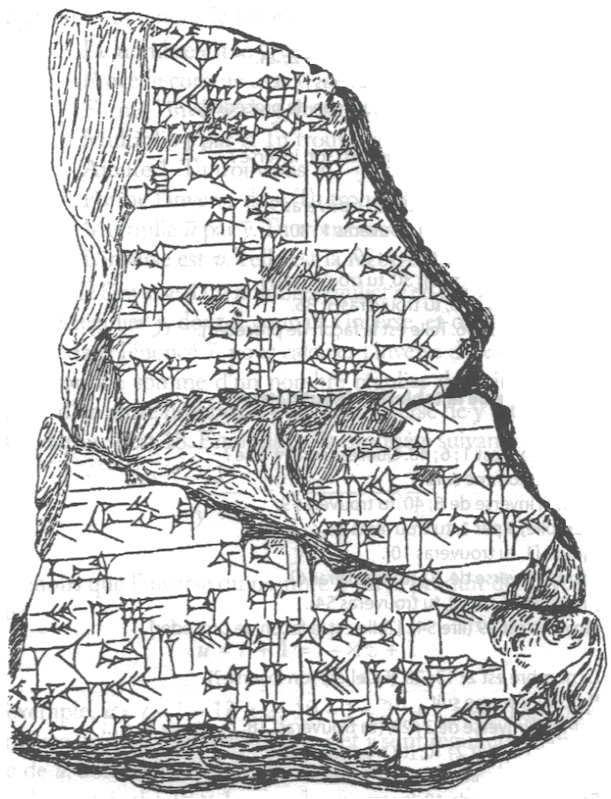
barra VAT 6505 do museu de Berlim

processo pelo exemplo (não existe ainda a noção de variável)
dos vários exemplos extraí-se o método:

O número é x, Qual é o seu inverso?
Procede assim.
Forma o inverso de y, obténs $\bar{y}$.
Multiplica $\bar{y}$ por z. Obténs t.
Junta 1. Obténs u.
Forma o inverso de u. Obténs $\bar{u}$.
Multiplica $\bar{u}$ por $\bar{y}$. Obténs v.
O inverso é v.
Tal é a forma como proceder.

a ideia: expressar x como a soma de dois números (y e z) onde y tem um inverso conhecido

os exemplos expostos na barra procedem por recursividade (de uns, se calculam outros)



$$\frac{1}{x} = \frac{1}{y+z} = \frac{1}{\frac{1}{y} \times z + 1} \times \frac{1}{y}$$

os babilónios perceberam que os valores numéricos para os quais o inverso expressa-se numa sequência finita têm a forma

$$2^p 3^q 5^r$$

são os números sexagesimais ditos **regulares**

se a é regular, então $\frac{1}{a}$ também

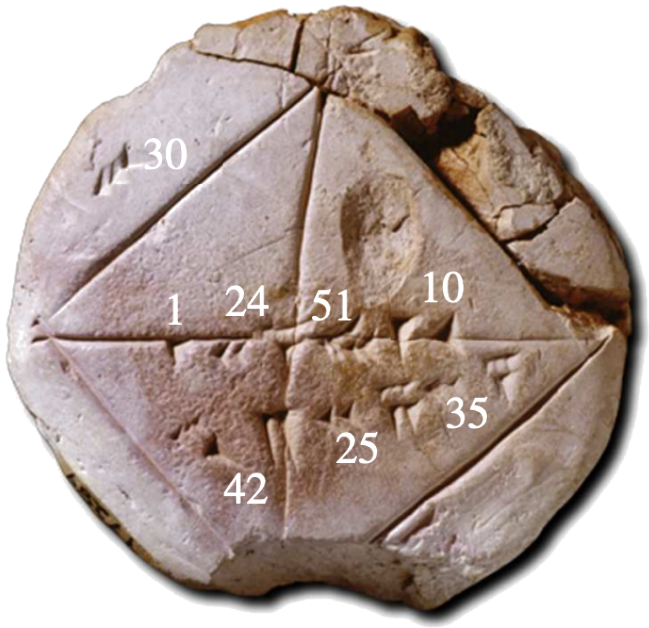
se a é inverso de b então $2a$ é inverso de $\frac{b}{2}$ $(...30b)$ e $\frac{a}{2}$ é inverso de $2b$

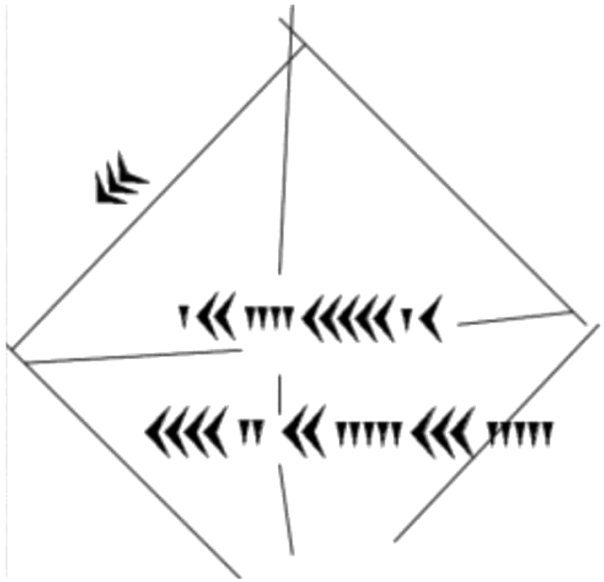
(generalizando ... $(a, b) \rightarrow (2a, 30b)$, $(2a, 20b)$, $(5a, 12b)$, $(6a, 10b)$)

há registo de barras de argila que listam os pares de inversos usando este método iterativo a partir de $(1, 1)$ de uma forma ordenada: primeiro registo de um cálculo humano que **demoraria mais de um segundo** num computador atual

a barra de argila YBC 7289: um TPC

esta barra de argila que aparenta ser o resultado de um exercício (data: entre 1900 e 1600 AC)
onde é calculada/aproximada $\sqrt{2}$ como comprimento da hipotenusa





mais de 1000 anos antes de Pitágoras!

método de babilónia ou de Herão de Alexandria
(relatado por este matemático, sec. I DC, em Métrica, vol I, 8 - descoberto só em 1896 da nossa era)

> *Porque os 720 não tem lado expressível, tomaremos como lado com uma muito pequena diferença. Porque o quadrado o mais vizinho é 729 e tem por 27 como lado, divide os 720 por 27 : resulta 26 e dois terços. Junta os 27 : resulta 53 e dois terços. Destes, a metade : resulta 26 2' 3'. O lado aproximado de 720 será então 26 2' 3'. De facto 26 2' 3' por ele próprio dá 720 36', por forma que a diferença é uma 36-ésima parte de unidade. Se queremos que a diferença seja produzida por uma parte ainda menor do que 36', no lugar de 729, utilizaremos os 720 36' agora encontrados e, retomando os mesmos passos, encontraremos uma diferença muito menor do que 36'*

-

Herão de Alexandria, Métrica, vol. I, 8

calcular $\sqrt{a}$ com $a \in \mathbb{R}^+$

$$\forall x \in \mathbb{N} \quad x_{n+1} = \frac{x_n + \frac{a}{x_n}}{2}$$

escolher x_0 *perto* do valor pretendido
com $a = 2$ e $x_0 = 1$ temos

$$\begin{array}{lcccl}
x_1 & = & \frac{1+2}{2} & = & \frac{3}{2} & = & 1,5 \\
x_2 & = & \frac{\frac{3}{2}+\frac{4}{3}}{2} & = & \frac{17}{12} & = & 1,4166\ldots \\
x_3 & = & \frac{\frac{17}{12}+\frac{24}{17}}{2} & = & \frac{577}{408} & = & 1,4142156\ldots \\
x_4 & = & \frac{\frac{577}{408}+\frac{816}{577}}{2} & = & \frac{665857}{470832} & = & 1,4142135623745\ldots \\
x_5 & = & \ldots & = & \frac{886710888897}{627013566048} & = & 1,41421356237309504880168896\ldots
\end{array}$$

matemática desta época espantosamente complexa, expressa como métodos de cálculos que mais tarde designaremos por algoritmos

vimos exemplos onde é claro que já conheciam o processo de **cálculo por recursividade**
encontramos ainda em barras descobertas **pérolas** como :

... guarda estes números na tua cabeça, toma o primeiro e troca-o por ele mais um...
noção de memória e de gestão desta como uma pilha

... troca a soma do comprimento com a largura por 30 vezes ela própria ...

noção de atribuição (x:=x/2)

A soma do comprimento e da largura é igual a área. Procede assim...

noção de precondição!

não falaremos aqui dos registos egípcios, da matemática chinesa, hindu, maya, inca, grega e romana, nem da dominância medieval da matemática árabe (e dos seus *métodos computacionais*)

passamos sob silêncio o contributo precoce de Eratósteno, Arquimedes, Ptolemeu, Diofante etc...

citemos só um caso que pelo seu impacto histórico merece o destaque: o algoritmo de Euclides

Euclides. Elementos VII.2 (300 AC) (formulação adaptada)

Dados dois segmentos AB e CD (com AB>CD),
retira-se CD de AB tantas vezes quanto possível.

Se não houver resto, então CD é a máxima medida comum.

Se se obtém um resto EF, este é menor que CD e podemos repetir o processo:

retira-se EF tantas vezes quanto possível de CD.

Se no final não restar nada, EF é a máxima medida comum.

No caso contrário obtém-se um novo resíduo GH menor que EF.

O processo repete-se até não haver nenhum resto.
O último resto obtido é a maior medida comum.

```
let rec mdc a b =  assert (a>b && b>=0);
   if b = 0 then a else mdc b (a mod b)
```

Da história ao nascimento registado

---

do ábaco às máquinas

ábaco

dispensa de apresentação alongada....

máquina rudimentar para expressar e registar valores numéricos de forma cómoda para as operações usuais para a qual se pretende usar estes valores

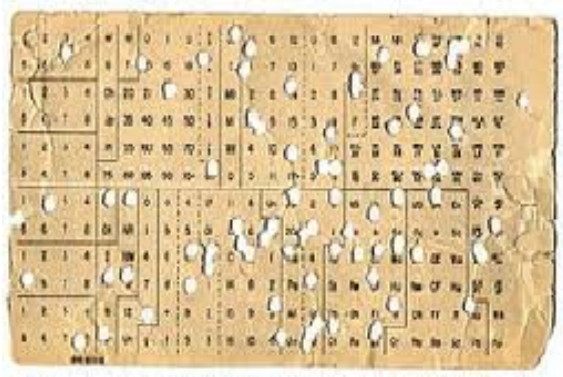
calcular **volumes, preços, impostos, taxas, juros, previsões financeiras** ...

indícios históricos permitam estimar que os primeiros ábacos apareceram **5500 AC** na Mesopotámia

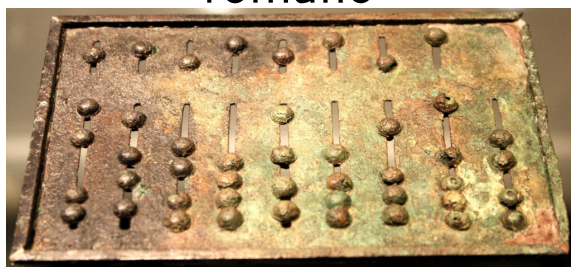
uso **transversal** em todas as grandes civilizações (os quipús são exemplos, mas também na civilização Azteca)

... precisa de um computador humano

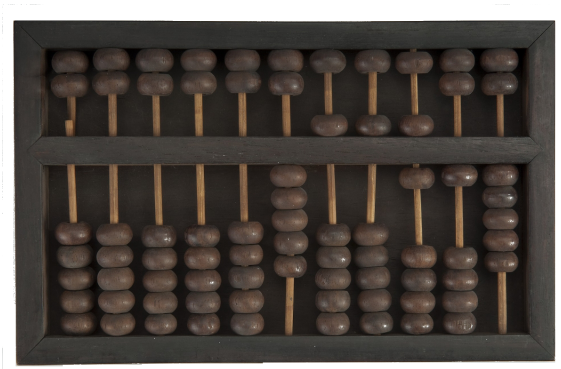
sumério



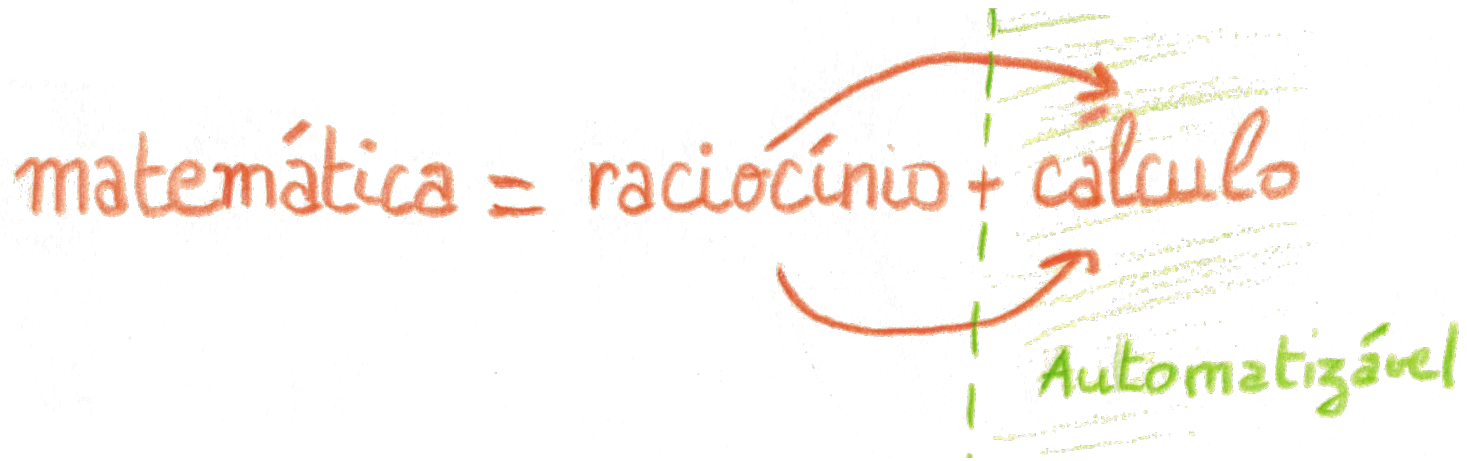
romano



chinês



em grande parte da história da matemática: o ábaco



até que ponto o próprio raciocínio não se pode levar ao cálculo?

a **Ciência** (matemática) : extender o conhecimento matemático, discriminar deste o que é exclusivamente criativo do que se pode reduzir ao cálculo, **expor estas reduções**

a **engenharia** : tentar mecanizar o que se sabe do cálculo

na nossa perspectiva (de quem usa e sabe o que são computadores modernos): o **ábaco calcula alguma coisa ?**

não... é uma memória mecánica bem organizada de auxílio à computação

assim, depois do renascimento matemático, o ábaco é uma obra de engenharia sobrevalorizada tendo em conta o conhecimento matemático acumulado

é desta constatação (e da reação dos grandes matemáticos e engenheiros da nossa história) de que brevemente falaremos

instrumento mecánico cujos primeiros vestígios remontam a 150 AC mas sofreu constantes melhorias ao longo da história e das civilizações

instrumento complexo que conseguia efectuar/auxiliar medidas complexas

utilizado para resolver problemas geométricos, geográficos, de astronomia

como também **ajudar no cálculo** da altura de um edifício ou a profundidade de um poço.

(réplica de um astrolábio do Irão do século XIII)

máquina de anticítera (...87 AC)

considerada como a primieira máquina de calcular analógica

descoberta no fundo do mar, perto da ilha de Creta em 1901

estraordináriamente complexa para os saberes que acreditavamos ser os da época

acredita-se que tenha por origem Alexandria, cidade que a História situa como um pólo importante da ciência da altura

a solidez da sua composição deixa pensar que se produziam tais máquina e que assim a sua origem possa ser bem anterior a sua datação (... Arquímedes (212 AC)?)

a máquina é constituída por um mecanismo complexo de engrenagens que permitam calcular/prever o tempo/calendário, eventos astronómicos (com base em cálculos sobre posições dos astros)

(site wikipédia para mais detalhes: (link))

Ramon Llull (Catalão, 1232-1315)

máquina lógica constituída por círculos concêntricos contendo palavras que, dispostas numa certa ordem, formavam perguntas e respectivas respostas

derrotar o pensamento herético (os mouros) convencendo-os

tentativa de codificação do raciocínio lógico (máquina de pensar)

engenhoso sistema de validação de (algumas) fórmulas lógicas proposicionais

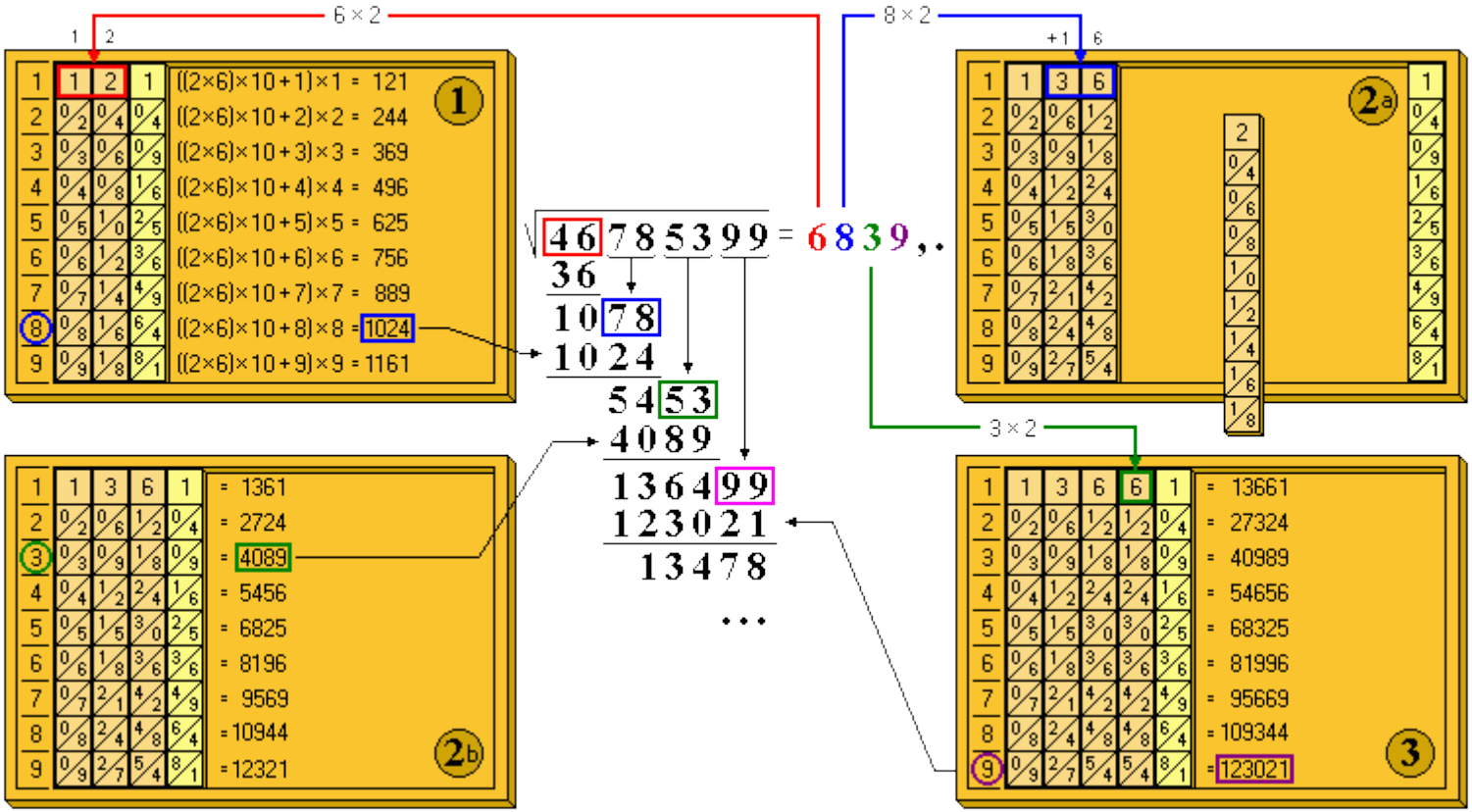
1617: os ossos de Napier

inventada por John Napier (Edimburgo, 1550-1617)

é uma **evolução do ábaco**, mas que consegue auxiliar o cálculo de **produtos, quocientes, potências, raizes quadradas**

(réplica datando do século XVIII)

1617: os ossos de Napier



(fonte : wikipedia)

1623: Wilhelm Schickard e o relógio calculador

construiu o **relógio calculador**: uma máquina de calcular mecánica capaz de realizar as 4 operações aritméticas básicas $(+ \ - \ \times \ \div)$ com **números de 6 dígitos** e **indicação de overflow** (via sinal sonoro!)

usada pelo seu amigo Johannes Kepler

perdeu-se num incêndio

os esboços do relógio, perdidos até o século XIX, deram lugar a uma réplica construida em 1960

1623: Wilhelm Schickard e o relógio calculador

construiu o **relógio calculador**: uma máquina de calcular mecánica capaz de realizar as 4 operações aritméticas básicas $(+ \ - \ \times \ \div)$ com **números de 6 dígitos** e **indicação de overflow** (via sinal sonoro!)

usada pelo seu amigo Johannes Kepler

perdeu-se num incêndio

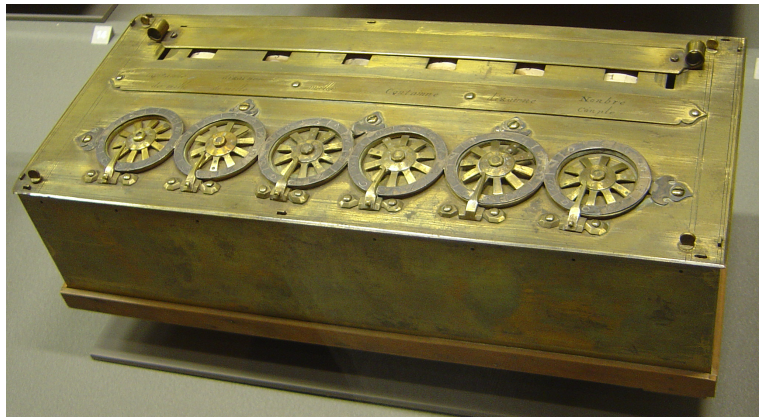
os esboços do relógio, perdidos até o século XIX, deram lugar a uma réplica construida em 1960

1642: Blaise Pascal e a Pascaline

Blaise Pascal, na altura com 19 anos, quis aliviar o trabalho contabilístico do pai (alto responsável da monarquia francesa afeto aos assuntos fiscais)

a pascaline realiza **adições e subtracções mecanicamente**, e realizar **multiplicações e divisões via repetição**

revolucionou a mecanização dos cálculos: a mecánica é exemplar e não há interpretações ou cálculos pendentes a realizar pelo utilizador, das entradas obtém-se o resultado final



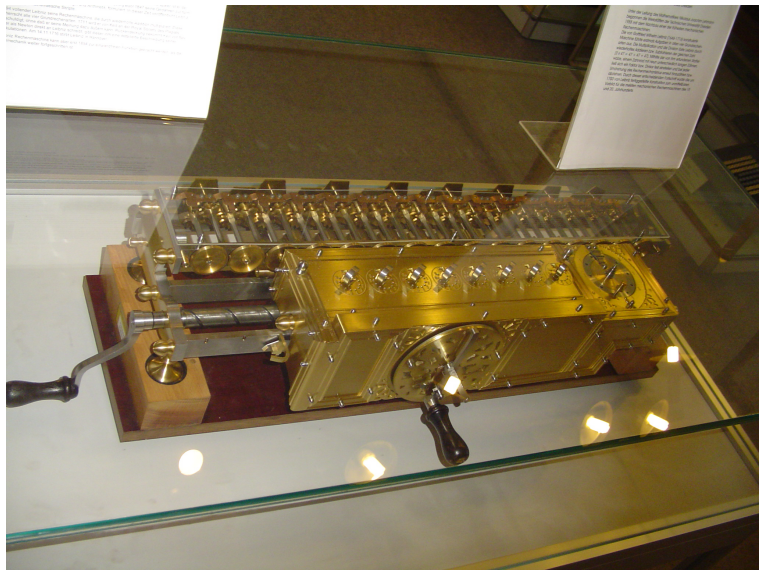
verdadeira obra de engenharia, foi possível replicá-la para uso comercial

a linguagem de programação pascal honra, pelo seu nome, este cientista

1673: Leibniz

Gottfried Wilhelm Leibniz (leipzig, 1646-1716)

em 1666 (com 20 anos) teoriza sobre uma máquina de cálculo universal, sonho de David Hilbert (perto de 250 anos mais tarde) a **calculus ratiocinator**



propõe em 1673 uma melhoria à pascaline por forma a executar **automaticamente** também **multiplicações e divisões**

... até a produção industrial de calculadoras mecânicas

Arithmomètre de Thomas de Colmar, 1820

Arithmometer de Odhner, 1873 - melhorias com engranagens

Comptometer de Dorr E. Felt, 1887

Da história ao nascimento registado

---

das calculadoras aos computadores programáveis

Joseph-Marie Jacquard

Joseph-Marie Jacquard (Lyon, 7 de julho de 1752 - Oullins, 7 de agosto de 1834)

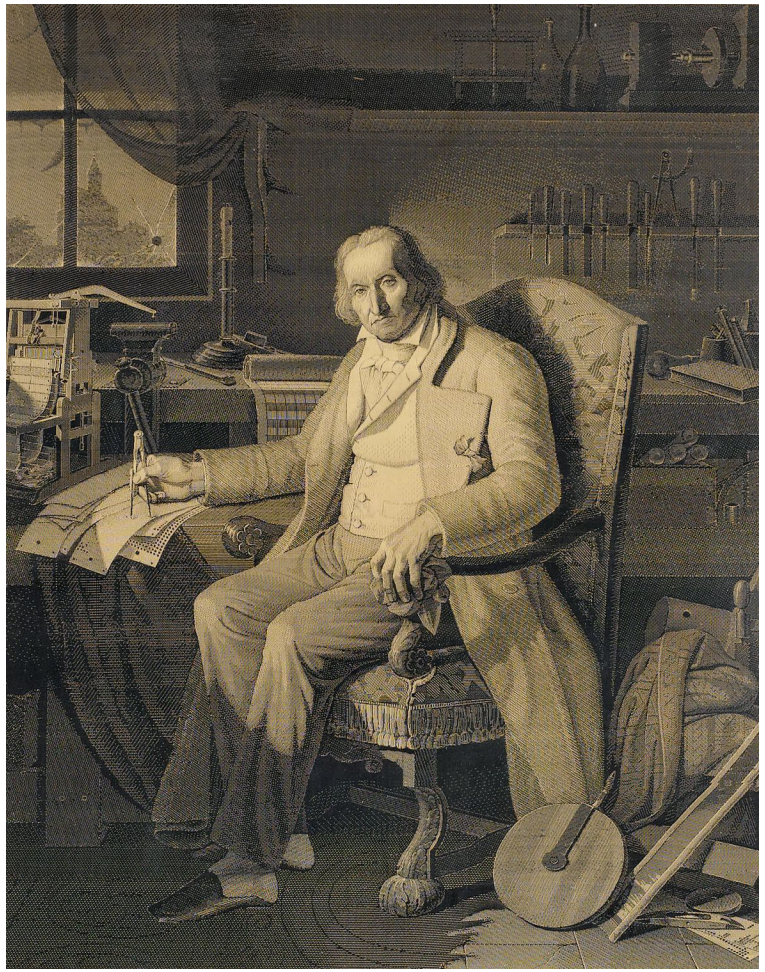
autodidacta em mecánica, trabalha jovem numa fábrica de tecelagem (ligado ao meio familiar)

trabalho muito duro, saúde frágil tenta mais tarde perceber, com o seu conhecimento técnico, como melhorar esta tarefa laboral e a vida dos seus operadores

desenha uma máquina para realizar rendas e bordados (de forma sistemática)

é com base com esta máquina que constrói o tear Jacquard

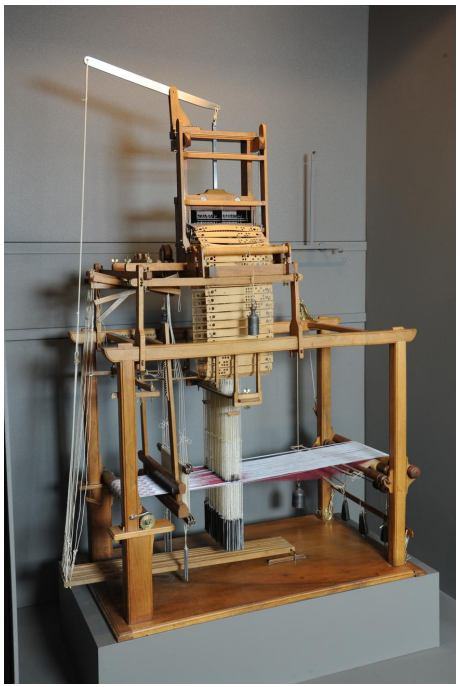
contribuiu à revolução industrial a acontecer, curiosamente os operários acharam que esta máquina iria destruir o emprego deles

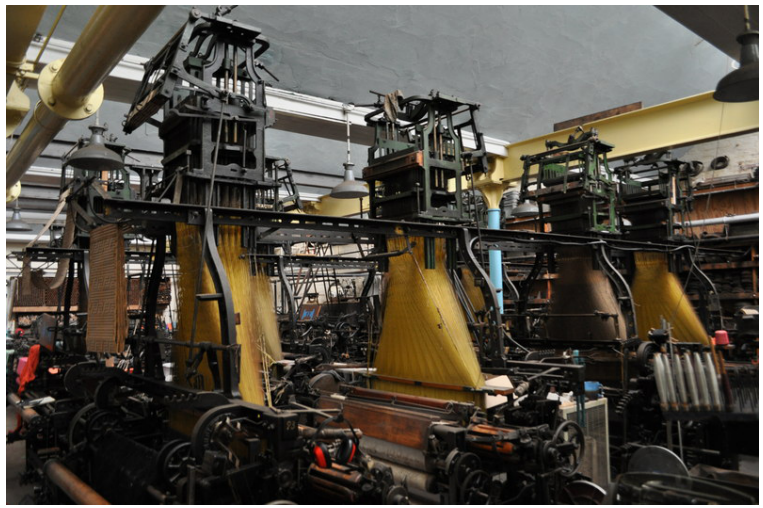


1801 - tear Jacquard

influência inegável sobre o que virá a ser o **computador programável**... ou o realejo

cartão perfurado que indica ao tear que **sequência de acções** (fios por puxar para baixo, fios por puxar para cima) realizar **mecanicamente**(sem intervenção humana) no processo de tecelagem para **produzir tecido** com determinados motivos





Charles Babbage

Charles Babbage (Londres, 1791 - 1871), brilhante engenheiro, pioneiro no desenho e engenharia de computadores

as infelicidades do percurso dele levaram-no a

falhar o lugar na História que o A. Turing mais tarde preencheu:

não viu concretizada a máquina que poderia ter sido de facto o primeiro computador desenhado e construido!

surpreendentemente teve um percurso que o liga muito ao de A. Turing:

também foi um elemento essencial na História Britânica: ajudou à criptanálise durante o esforço de guerra do Reino Unido (quebrou o chiffre de Vigenère)

Charles Babbage e a máquina diferencial

em 1786, J.H. Müller (engenheiro alemão no Canadá) concebe a ideia de uma máquina diferencial

não consegue fundos para a construir

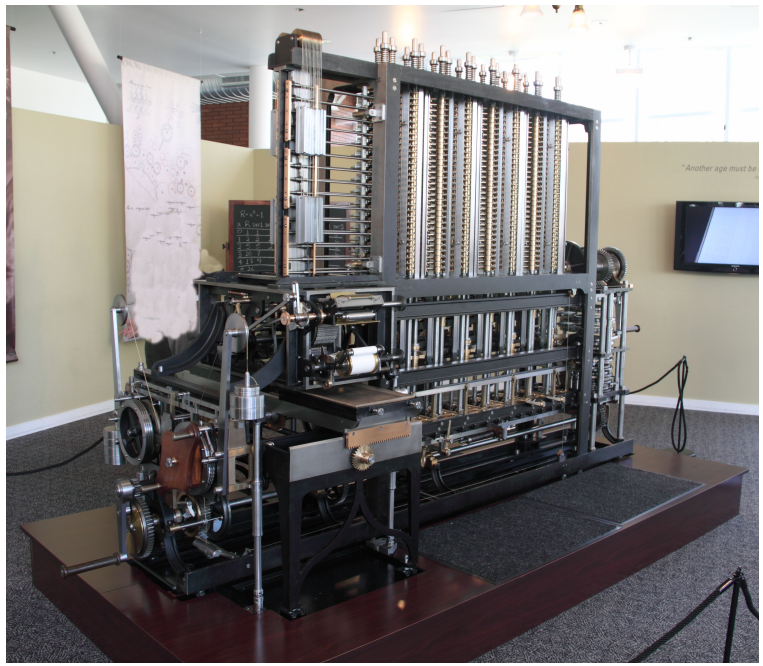
em 1822, C.Babbage desenhou uma evolução desta máquina e consegue convencer o Reino Unido em financiar a construção

nesta altura, em plena revolução industrial, os estados idealizam a posição estratégica (e a poupança gerada) em ter poderio computacional (mecânico)

a máquina construida era capaz de resolver equações polinomiais através de diferenças

esta máquina conseguia igualmente construir tabelas de logaritmos.

... mas precisava de alterações



(a máquina diferencial)

Charles Babbage e a máquina analítica

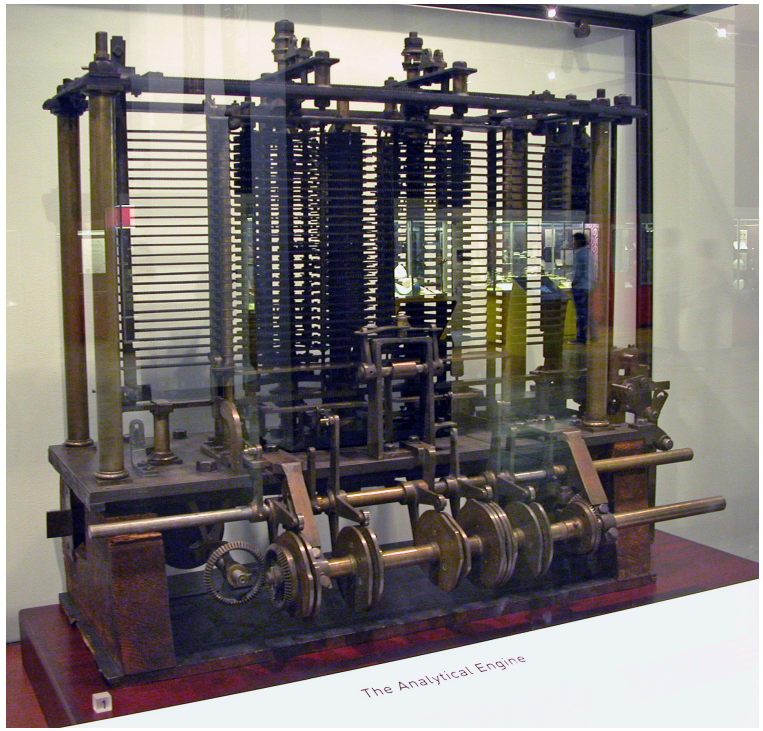
em 1833, C. Babbage percebeu como melhorar a máquina diferencial e propor um mecanismo de computação mais geral: a **máquina analítica**

prefigurou o que viriam a ser os computadores

incluía um **dispositivo de leitura de cartões perfurados**

C. Babbage desinteressou-se da máquina diferencial e dedicou todos os seus esforços e recursos, em vão, na construção da máquina analítica

os conhecimento de engenharia da altura não eram suficientes para conseguir concretizar a máquina (e o Reino Unido recusou financiar a investigação e a construção)



(protótipo inacabado da máquina analítica)

Ada Lovelace

Ada Augusta King, Condessa de Lovelace (nascida Byron, Reino unido, 1815 - 1852)

filha de Lord Byron, mas educada pela mãe que lhe deu uma formação matemática como forma de fugir ao destino do pai

matemática experiente, conheceu o C. Babbage numa exposição da máquina diferencial e começou a trocar correspondência com ele sobre potenciais usos da máquina analítca

ajudou o C. Babbage na divulgação científica do projecto dele (fatuamente, uma tradução que ultrapassou claramente o seu propósito e ficou para a posteridade)

a linguagem de programação ADA honra, pelo seu nome, esta cientista

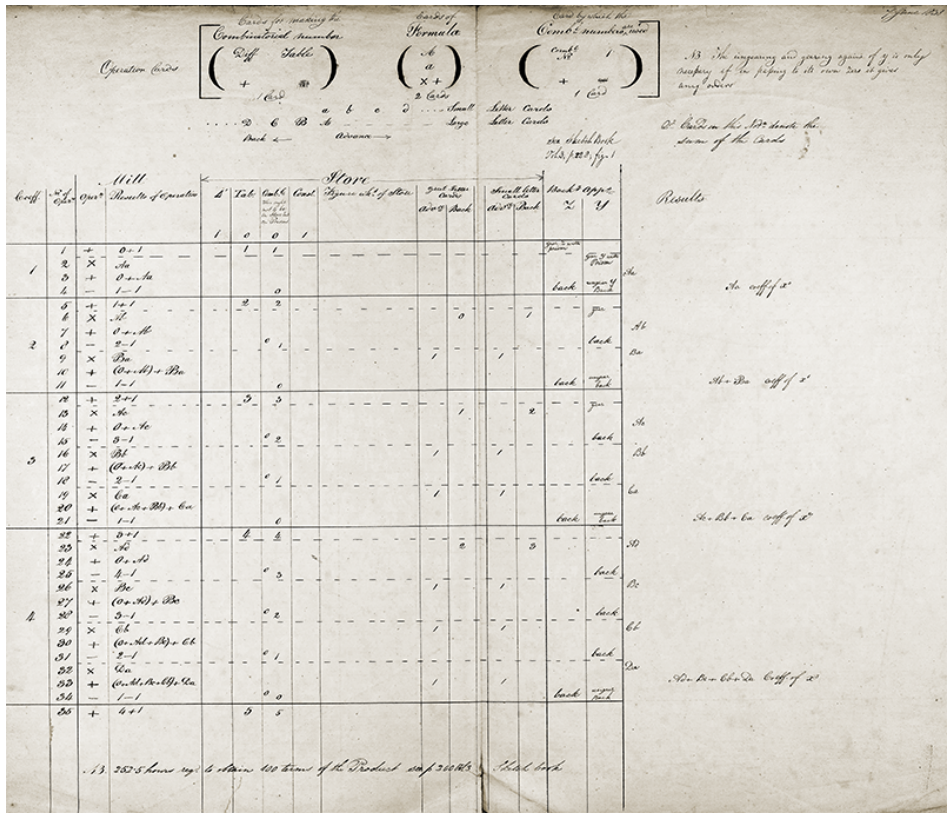
trocou **correspondência** com o C. Babbage sobre o uso da máquina analítica para que esta seja configurada para produzir em saída elementos da **sequência de Bernouilli**

desta troca com o Babbage nasceu o que se considera ser **o primeiro programa**

(para ser totalmente executado mecanicamente sem intervenção humana por um computador que não existia!)

preparado para tomar a forma de um cartão perfurado

programa, e não algoritmo!



The Analytical engine weaves algebraic patterns just as the Jacquard loom weaves flowers and leaves *(Ada Lovelace)*

no final do século XIX e no início do século XX nascem várias iniciativas empresariais, como a futura IBM (Hollerith, 1896 Tabulation Machine Company e em 1911 Computing-Tabulating-Recording Company) que constroem máquinas electromecânicas

em 1938, nasce a HP

1939 - John Atanassoff e Clifford Berry: protótipo **ABC**, que é considerado como o primeiro computador digital

John von Neumann (Budapeste, 1903 - Washington, D.C., 1957) matemático, lógico, físico, pioneiro da computação participou no projecto Manhattan

1945 - John von Neumann propõe o seu modelo de arquitectura interna de uma máquina universal (computador)

1946 - von Neumann participou na construção do **ENIAC**, último computador eléctrico programável (baseado na arquitectura de Von Neumann)

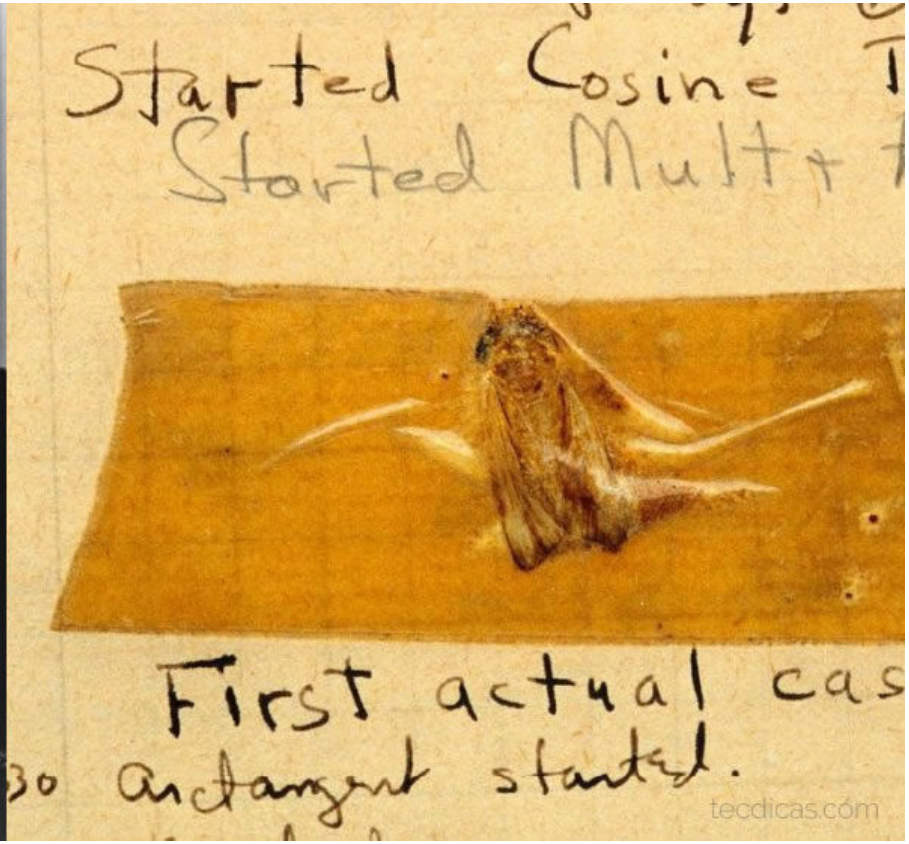
1949 - participou também na construção do **EDVAC**, o primeiro computador electrónico baseado na arquitectura de Von Neumann.

Grace Murray Hopper (USA, 1906 - 1992), pioneira da programação e das linguagens de programação

se Corrado Böhm idealizou (na sua tese de doutoramento) o primeiro compilador, em 1951, a Grace Hopper foi a primeira a implementar um **concretamente** em 1952
(para uma linguagem que viria a evoluir para o COBOL)

da sua actividade resultou a designação de **bug** para os erros em programas

yes, that's a bug



Da história ao nascimento registado

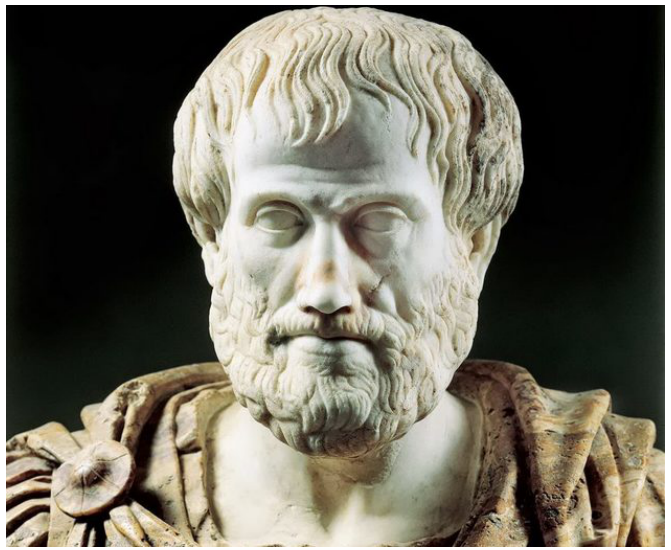
... e a informática nasceu

Aristóteles (Grécia 384 - 322 AC) : **a lógica deductiva**

viveu em Atenas onde definiu as básicas axiomáticas do raciocínio lógico

reductio ad absurdum, **modus ponens**, **modus tollens**... os silogismos...

reconhece-se que juntos com Platão e Sócrates tenha estabelecido as bases do que seria o conhecimento científico na Europa



Euclides (Alexandria 300 AC) : **a prova matemática**

deu as bases e estruturou a Matemática como esta seria entendida até ao século XIX

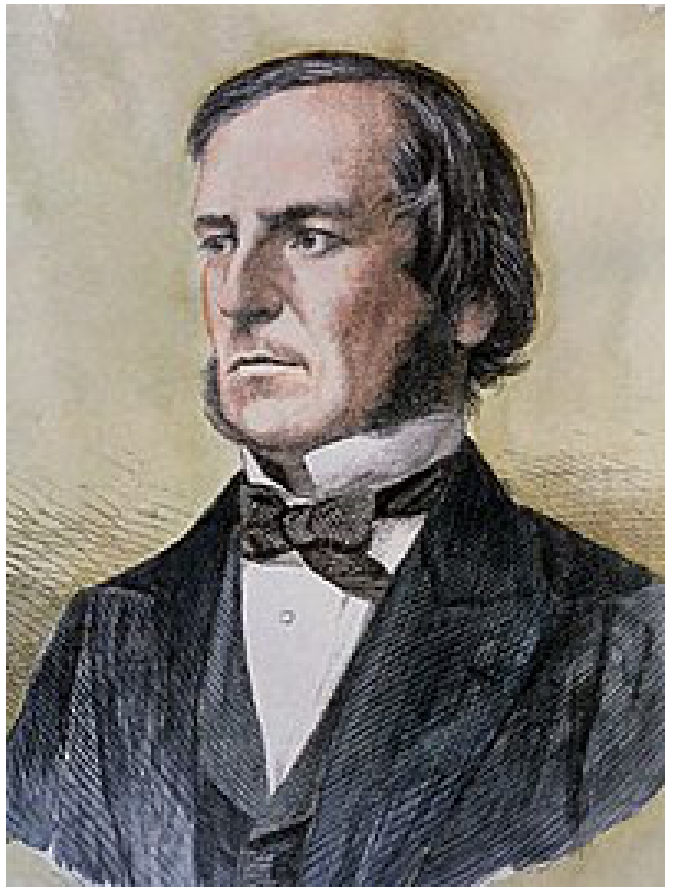
fundamentou-a com a sua formalização da Geometria, baseada em princípios matemático rigorosos

o seu livro "Os elementos" foi referência até ao século XIX

George Boole (Reino Unido, 1815 - 1864) : **álgebra de Boole**

em 1854, o lógico inglês publica o livro "The mathematical Analysis of logic" onde define os operadores lógicos baseados nos valores 0 (que significa falso) e 1 (verdade): os famosos operadores booleanos

criou assim a álgebra booleana, fundamental para o desenvolvimento da computação

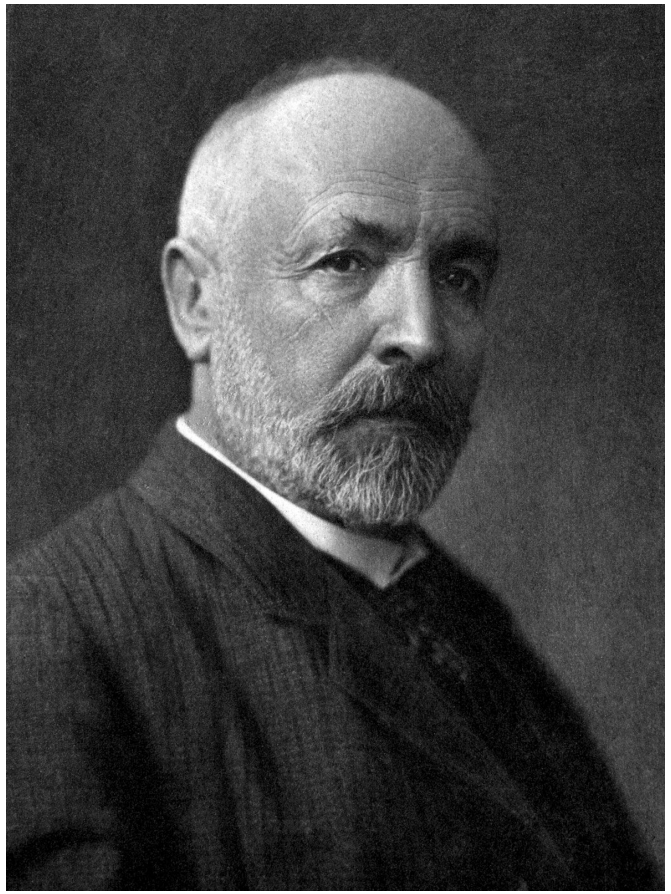


Georg Cantor (Russia 1845 - Alemanha 1918) : **a teoria dos conjuntos**

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 dele em 1926: "*Ninguém nos expulsará do paraíso que Cantor criou para nós*"



Gottlob Frege (Alemanha 1848 - 1925) : **a lógica predicativa e os seus quantificadores**

inventou noção de quantificadores da qual segue a Lógica de Primeira Ordem

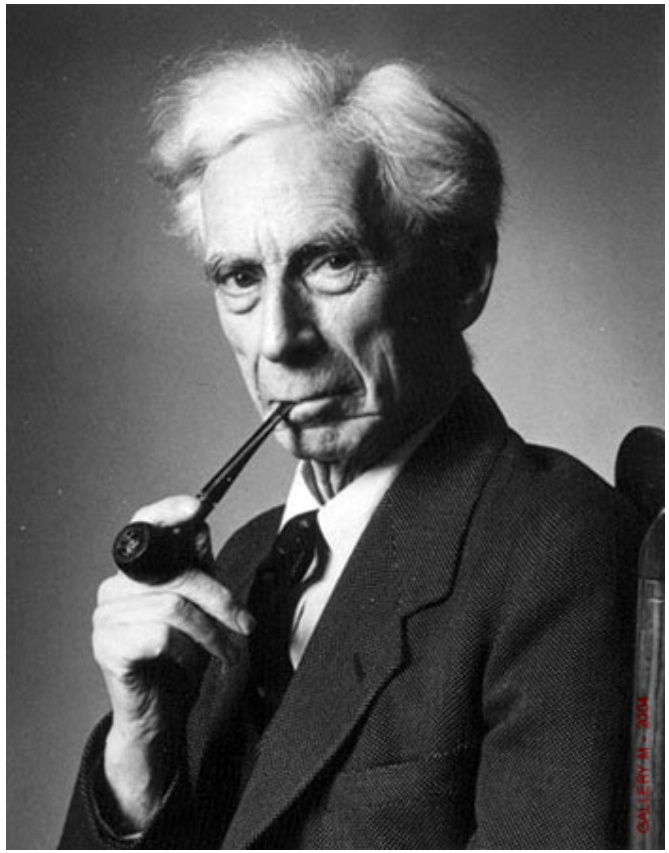
inventou a Lógica de Segunda Ordem

essa permitiu o tratamento rigoroso das noções de funções e variáveis, tendo formalizado a aritmética

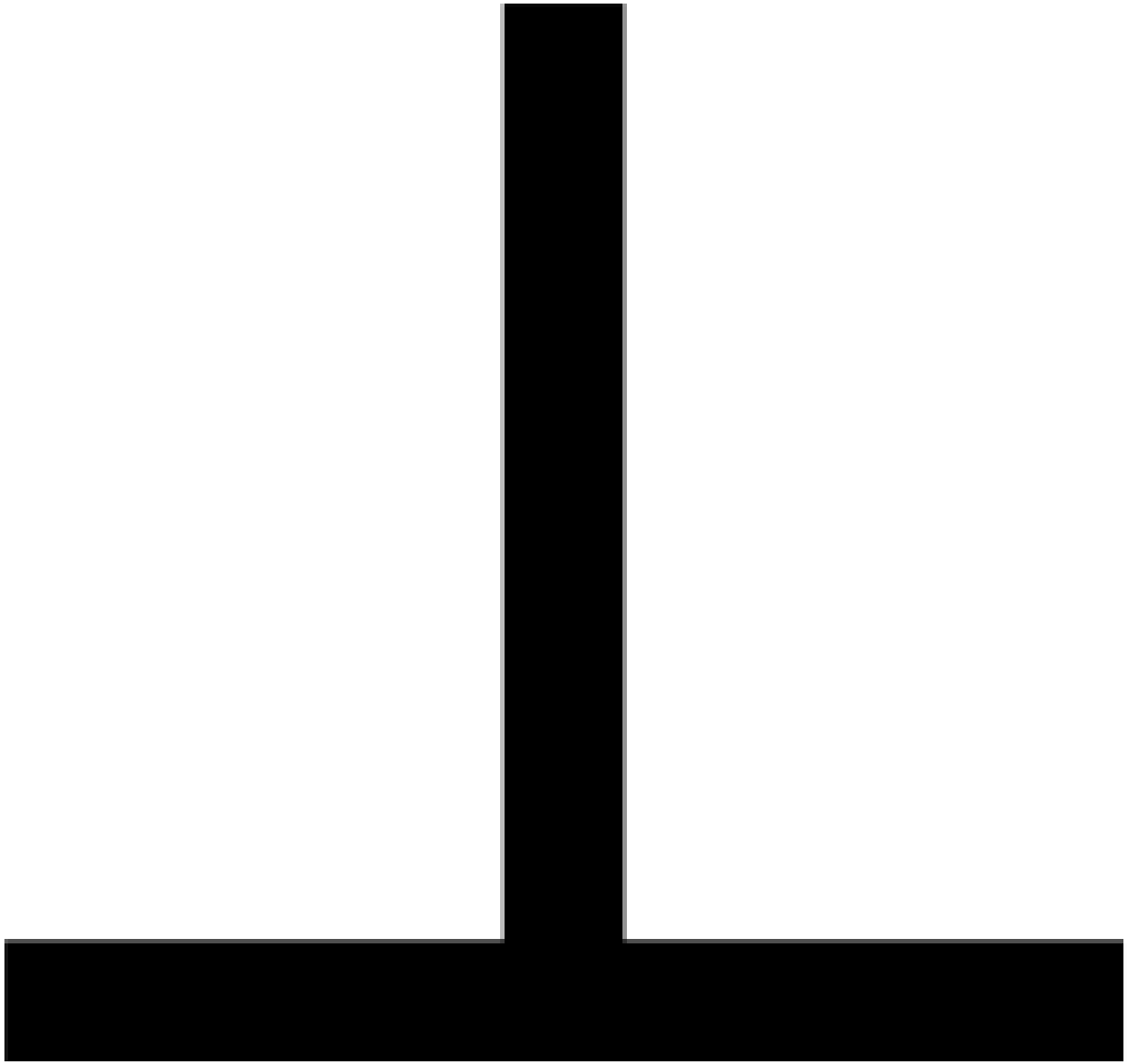
Bertrand Russell (Reino Unido, 1872 - 1970) : **a lógica como fundação para a matemática**

Escreveu, **Principia Mathematica** (com Alfred North Whitehead), 3 vols. uma **obra prima**

ali descreve como a matemática pode ser reduzida (i.e. fundamentada) a um conjunto pequeno de princípios fundamentais



lembrete: fruto proibido na matemática




BAM!

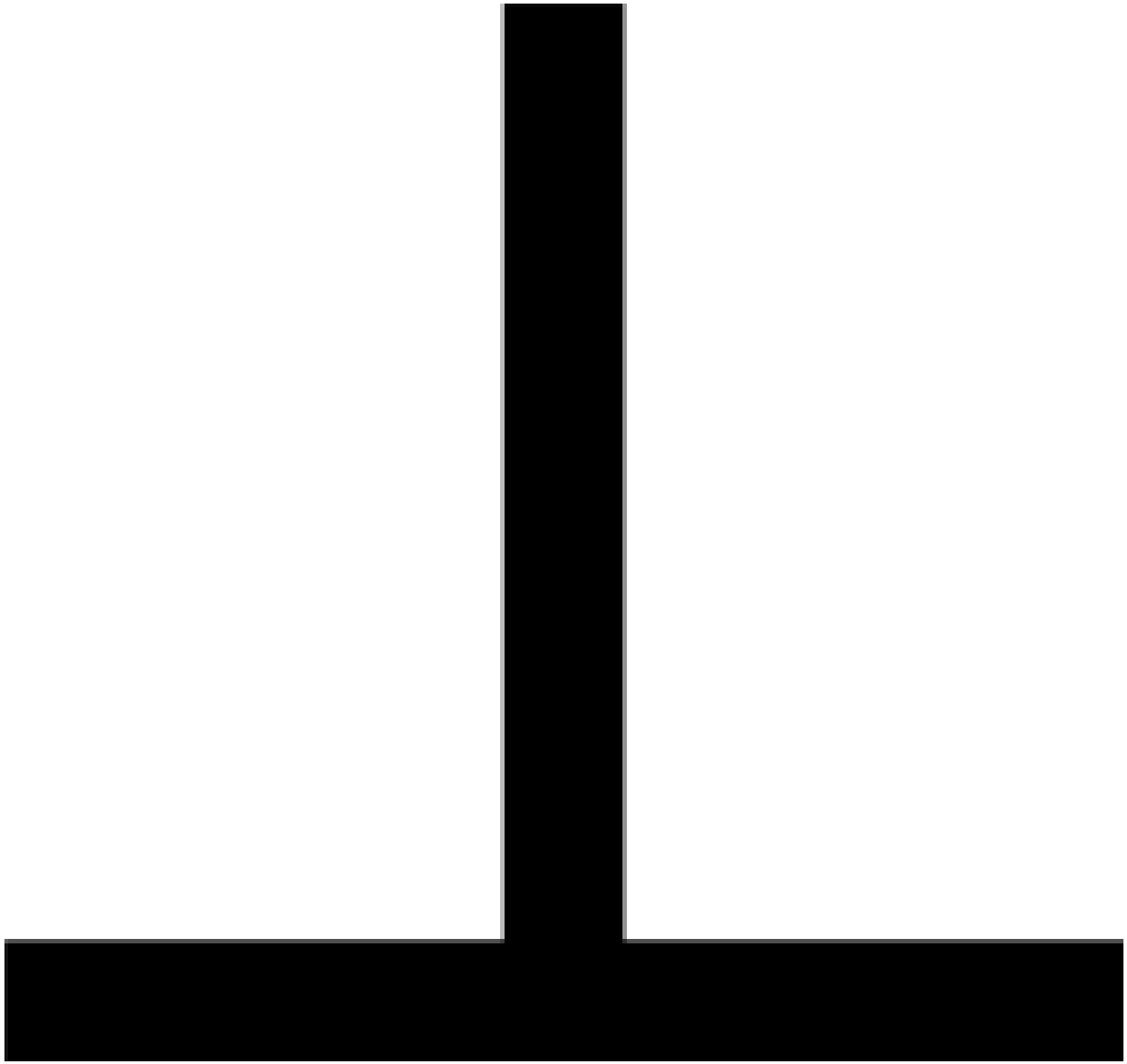
Epiménides, que era cretense (natural de Creta) disse:

> *"Todos os cretenses são mentirosos"*
> *(citado na Epístola de Paulo a Tito, Tito 1:12)*

uma variante:

> *Esta afirmação é falsa.*

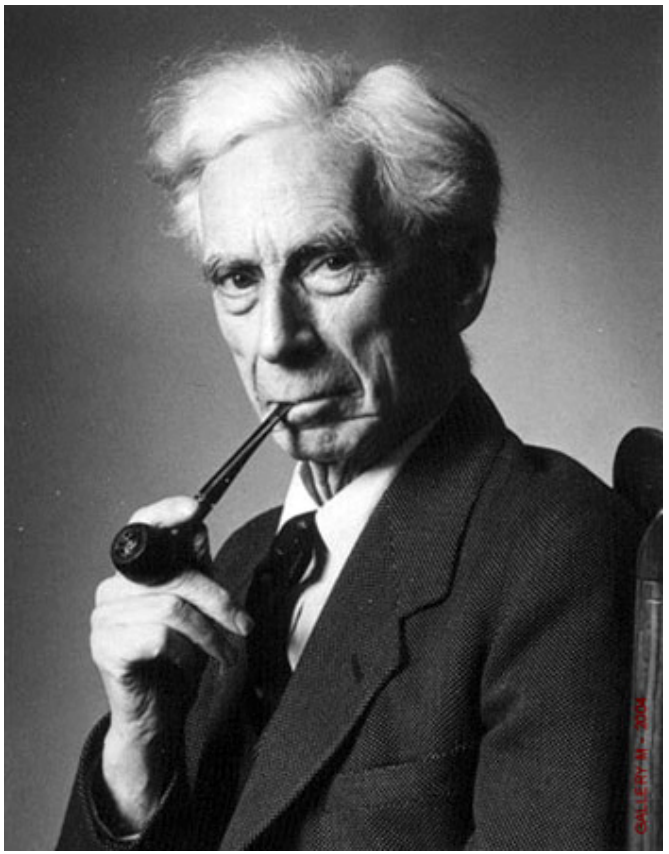
mas... então?... é verdade? não?...




BAM!

voltemos ao **Bertrand Russell**

Estudou os fundamentos da Matemática, em particular como o Fredge a formalizava e descobriu um paradoxo (poucos dias antes de Fredge querer publicar a obra dele)

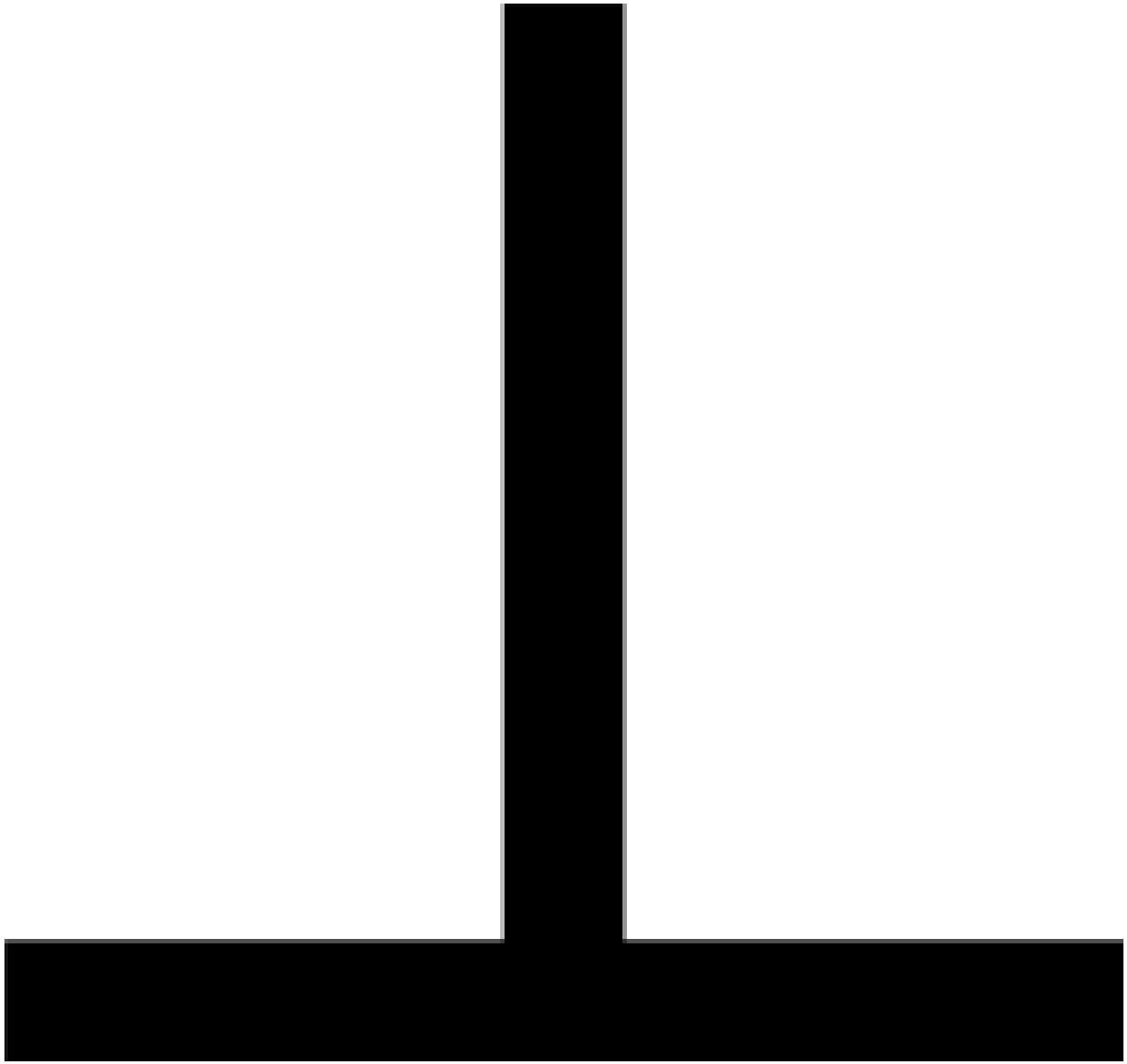


o conjunto M de todos os conjuntos que não se contém a si próprio, contém-se a si próprio?

$$M = \{A \mid A \notin A\}$$

analise por caso: se sim.... não se não... sim

mas... então?... não, sim...




BAM!

David Hilbert (Alemanha 1862 - 1934) : **podemos usar a lógica como fundação da matemática?**

pode a matemática se resumir totalmente ao cálculo?

definiu um programa (em 1900, e actualizado em 1928) de investigação par formular **toda** a Matemática com bases sólidas e assentes na lógica a

afirmou mesmo:

Temos que saber, viveremos a saber!

para sublinhar que a Matemática desvendaria toda a Verdade

David Hilbert (Alemanha 1862 - 1934) : **23 problemas para o futuro da matemática** - na perspectiva da época destacamos dois problemas

1900 - Décimo Problema. Q1: " Existirá um **método efectivo** para resolver qualquer equação diofantina ?' (*mas... o que é um método efectivo?*)

1928 - *entcheidungsproblem*, o Problema da Decisão. pode a Matemática se resumir totalmente ao cálculo (lógico)?
$\rightsquigarrow$ existirá um conjunto de axiomas tais que

- Q2: consistente
- Q3: completo
- Q4: decidível (i.e. **mecanizável**)

$\rightsquigarrow$ **Q5: existirá uma máquina que resolva com precisão todos os problemas matemáticos?**

Q1. **Não** ← obrigou à definição matemática de algoritmo - (1970 - Yuri Matiyasevich)

Q2. **Sim** Godel, Post, etc.

Q3. **Não** Godel, etc.

Q4. **Não** ← obrigou à definição matemática de algoritmo - Church, Turing, Kleene, Post, Rosser, Rice, etc.

Q5. **NÃO!** ... é privilégio dos matemáticos!

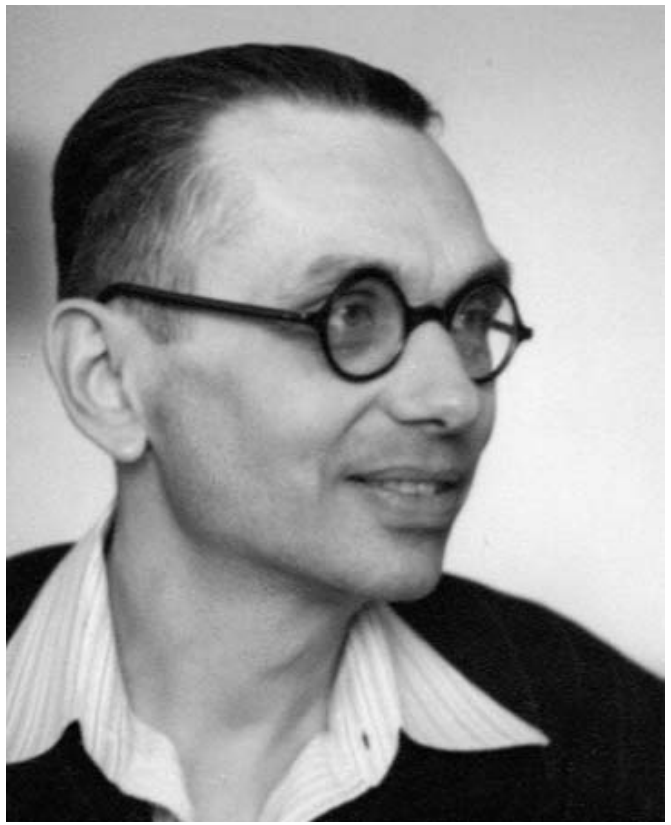
Kurt Gödel (Austria 1906 - USA 1978) : **a consistência e a (in)completude**

A. Einstein tinha nele e na Emmy Noether (e.g. Noetherian induction) uma admiração notável

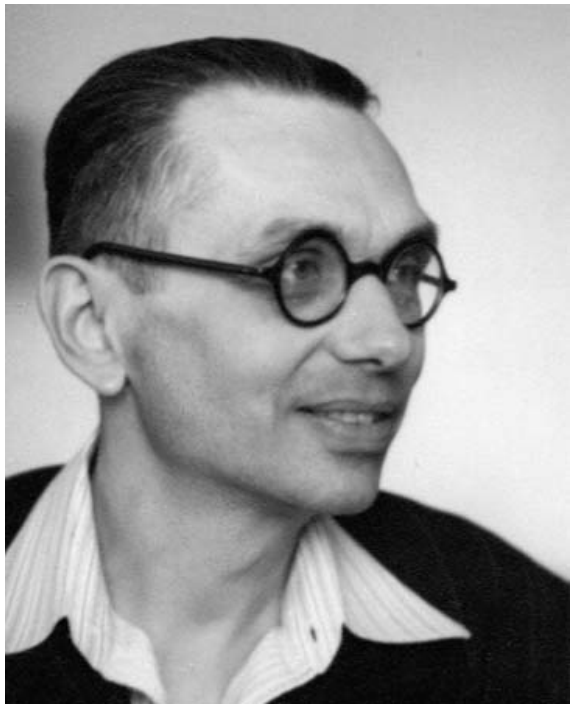
mostrou a **completude da lógica de primeira ordem** (no seu PhD, aos 24 anos)

e a **incompletude** de **qualquer** sistema axiomático para a **aritmética** (1931)

para a incompletude da aritmética, definiu o conceito de **números de Gödel**: um **compilador** das provas em aritmética para valores numéricos.... e assim exibiu que há necessariamente uma **instância do paradoxo dos mentirosos na aritmética**



- **incompletude** : num sistema formal axiomático **consistente** com **pelo menos** o poder da aritmética haverá sempre afirmações verdadeiras que não é possível demonstrar, em particular não pode provar a sua própria consistência
- **inconsistência** : um tal sistema, se é completo, não é consistente



anunciou os resultados num congresso em homenagem a David Hilbert

John von Neumann reagiu

está tudo acabado!

Alonzo Church (USA, 1903 - 1995) : **o nascimento das linguagens de programação... e da computação**

no seu esforço em fundamentar a matemática com uma **teoria axiomática das funções** matemáticas e do seu modelo de cálculo - o **cálculo lambda** - definiu o que era **computável e o que não era**

expressou com este meio a **indecidibilidade** da matemática na sua generalidade (1936)

e , efeito lateral, o cálculo lambda é a **primeira linguagem de programação**

Λ : conjuntos dos programas (i.e. lambda termos)

sintaxe

$$x \in \mathcal{V} \implies x \in \Lambda$$

$$x \in \mathcal{V}, M \in \Lambda \implies (\lambda x.M) \in \Lambda$$

$$M, N \in \Lambda \implies (M\ N) \in \Lambda$$

execução

$$(\lambda x.M)N \xrightarrow{\beta} M[N/x]$$

Alan M. Turing (Reino Unido, 1912 - 1952)

nos dias de hoje, dispensa quase de apresentação...

descodificou, com a ajuda de um engenho desenhado por ele (**The Bombe**) o **Enigma** (sistema de cifra alemã)

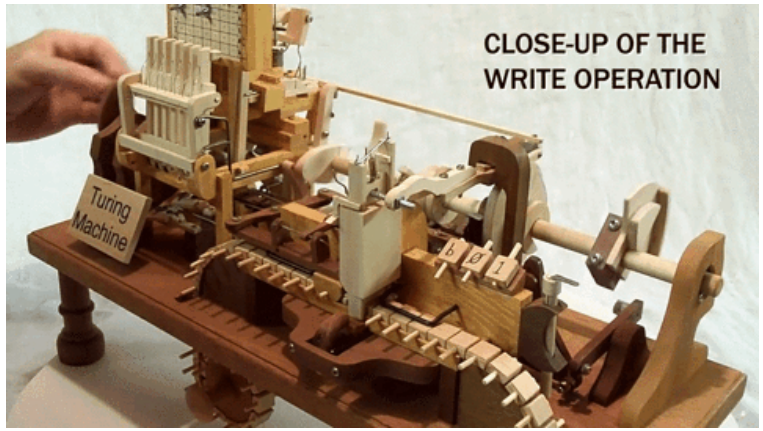
> *um homem só não ganha uma guerra, mas sem Turing a Grã-Bretanha teria provavelmente perdido a segunda guerra mundial (Winston Churchill)*

foi também pioneiro na criação da disciplina de Inteligência Artificial

sob a orientação do Alonzo Church, com 24 anos, o Turing desenvolveu a Máquina de Turing

a ideia era estudar o *entcheidungsproblem*

que mecanismo geral e uniforme podemos desenhar para efectuar **um** qualquer **cálculo**?
$\Longrightarrow$ Máquina de Turing



que mecanismo geral podemos desenhar para efectuar **qualquer cálculo**?
$\Longrightarrow$ Máquina Universal de Turing
(um máquina de Turing que aceita em entrada uma máquina de Turing)

uma aspecto notável: a ideia de que **programa = dados**

é um algoritmo qualquer processo que se pode expressar por (reduzir a) uma máquina de Turing ou lambda termo e cuja execução é finita

(Toda a função algoritmicamente calculável é computável por uma Máquina de Turing)

1936.... e com isso nasceu a informática!

uma curiosidade: Church teve uma visão **software** e Turing **hardware** da noção de computação

é uma tese, **não é** nem uma **definição** nem um **teorema**

então, qual o sustento?

- **Argumentos intuitivos** : nunca foi exibido um exemplo de tratamento efectivo que não pudesse ser implementado por uma máquina de Turing
- **Enriquecimento dos modelos da computação** : foram propostas várias extensões às máquinas de Turing (fitas duplas, mais movimentos da cabeça de leitura/escrita, etc)

  ficou demonstrado que não traziam poder expressivo suplementar em comparação com o modelo original. I.e.:
  - podia-se simular as extensões propostas no próprio modelo original
  - nenhuma extensão podia realizar cálculos que o modelo original não pudesse
- **Equivalência dos modelos computacionais** : modelos alternativos foram propostos (Post, Gödel-Kleene, Markov, Lambek-Minsky, Scott, etc.)

  ficou demonstrado que tinham exactamente (nem **nem mais, nem menos**) o mesmo poder expressivo

Da história ao nascimento registado

---

o problema da paragem e a Indecidabilidade

admitamos a existência de uma função $termina : programa \rightarrow bool$ tal que,

para um programa *qualquer* p, $termina(p) = true$ se para todas entrada possíveis de p, p **termina**

$termina(p) = false$ senão

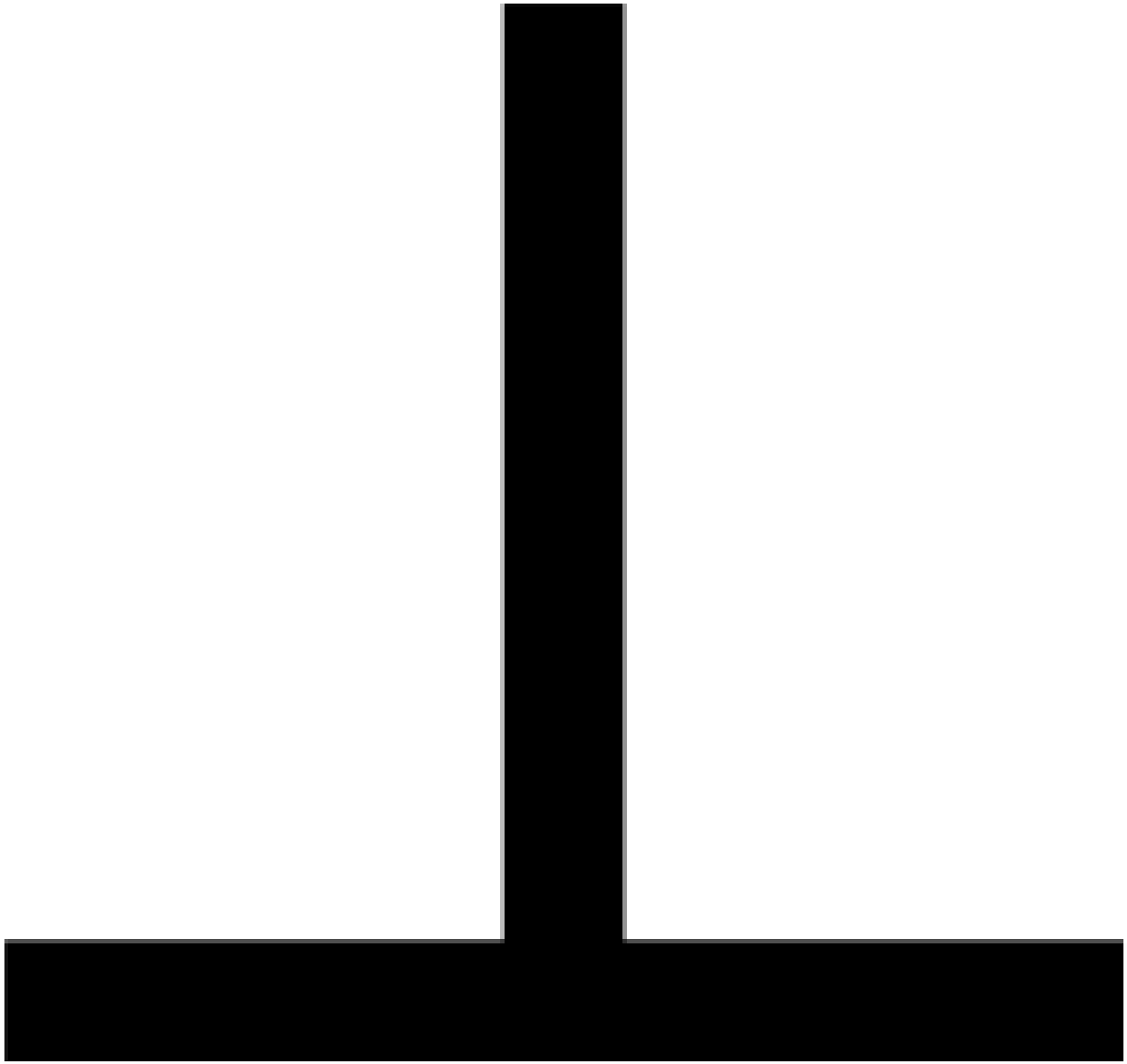
seja *misterio* o programa $if\ termina(misterio)\ then\ misterio\ else\ true$

qual é o resultado de $termina(misterio)$? depende claramente de *misterio*

se *misterio* **não termina**, então *termina*(*misterio*) devolve *false* logo *misterio* devolve *true* logo **termina**!

se *misterio* **termina**, então *termina*(*misterio*) devolve *true* logo *misterio* chama-se recursivamente e logo entra em recursividade infinita, logo **não termina**!

mas... então?... não, sim...




BAM!

o programa *termina* não pode existir

(redescobrimos o paradoxo do mentiroso)

a indecidibilidade é uma barreira inquebrável!

mas há situações **semi-satisfatórias**

... programas que quando terminam dão o resultado certo (sabem dizer **sim**), mas que podem eventualmente entrar em "**ciclo infinito**" na tentativa de dizer **não**

ou seja, quando há demora, não sabemos se é por estar a demorar muito a produzir uma solução ou porque não há simplesmente solução positiva: **semi-decisão**

epílogo

... é de contar que o meu computador sabe

contar é estabelecer uma relação unívoca com os naturais:

dado um conjunto arbitrário de elementos, uma sua enumeração é uma relação um-para-um entre cada elemento do conjunto e uma sequência crescente ininterrupta de naturais, a partir do 0

o número de elementos de um conjunto pode ser determinado desta forma?

0 é o número de elementos do conjunto vazio $\emptyset$;

1 é o número de elementos de qualquer conjunto singular (e.g $\{\emptyset\}$);

2 é o número de elementos de qualquer conjunto com duas entidades distintas (e.g $\{\emptyset, \{\emptyset\}\}$);

etc.

ou seja, o número de elementos do conjunto é o máximo da enumeração desses elementos

chamemos-lhe **cardinal**

mas é possível contar o número de elementos de qualquer conjunto?
como proceder se o conjunto não é finito?

em geral, dois conjuntos têm o mesmo número de elementos se existe uma bijecção entre os seus elementos

este critério chama-se **equipotência**

seja então $\aleph_0$ o número de elementos dos naturais

estabelecendo bijecções entre os conjuntos que pretendemos contar e os naturais encontramos o seu número de elementos

ou seja, este critério permite determinar o cardinal de qualquer conjunto, mesmo não finito, certo?

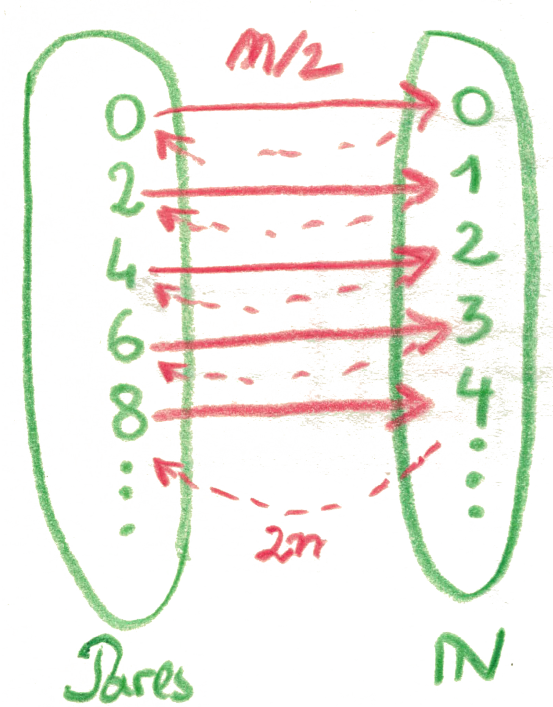
se o conjunto não é finito mas há uma bijecção com os naturais, esse conjunto tem o cardinal dos naturais

todo o conjunto equipotente a um subconjunto dos naturais é dito numerável (ou contável)

e se não há bijecção? analisaremos a questão mais tarde...

este critério da equipotência permite chegar a conclusões surpreendentes

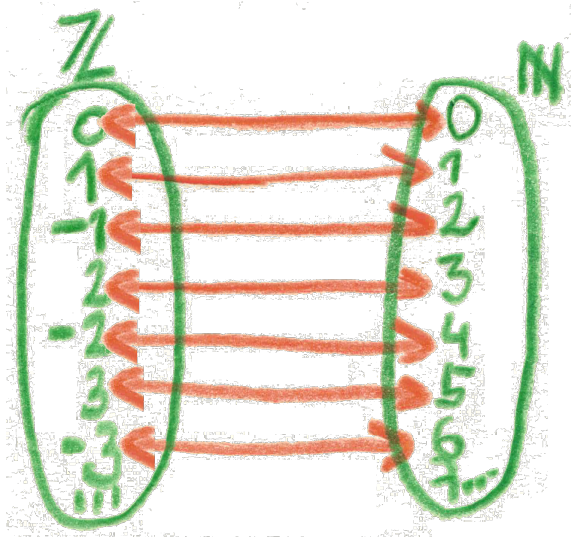
por exemplo, há tantos naturais pares como naturais!
seria de esperar que o cardinal do primeiro conjunto fosse metade do segundo, mas a função $f(n) = 2n$ é bijectiva!



claro que da mesma forma se conclui que o cardinal do conjunto dos naturais ímpares é também igual a $\aleph_0$

na verdade, qualquer subconjunto não finito dos naturais tem cardinal $\aleph_0$

e se tomarmos superconjuntos de naturais já obtemos cardinais superiores?
consideremos os inteiros ($\mathbb{Z}$)



a função "*em zig-zag*" que mapeia

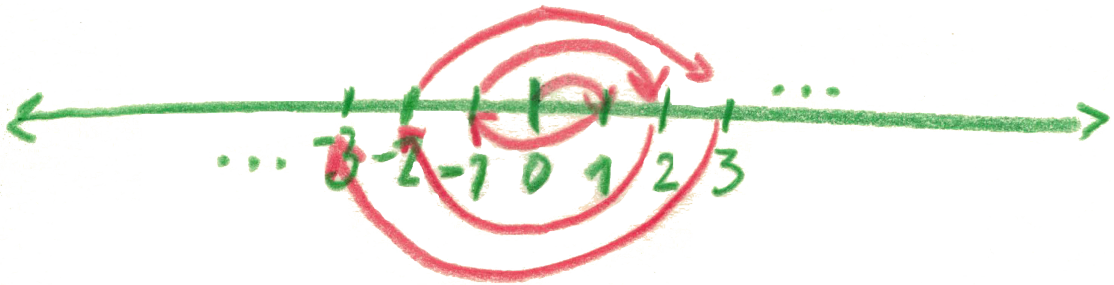
o zero dos inteiros no zero dos naturais

os inteiros positivos nos naturais ímpares

e os inteiros negativos nos naturais pares

é uma **bijecção**

logo, concluimos que o **cardinal dos inteiros** é também $\aleph_0$



E os racionais?

Se os virmos como pares de inteiros (considerando apenas fracções irredutíveis) e considerarmos a função que mapeia o par $(0,0)$ no 0, o $(1,0)$ no 1, o $(1,1)$ no 2, o $(0,1)$ no 3, o $(-1,1)$ no 4, o $(0,-1)$ no 5, etc,

temos uma função **bijectiva**!

logo, também os racionais têm cardinal $\aleph_0$, são numeráveis

ou seja **são tantos quantos os naturais**

na realidade, qualquer conjunto de sequências finitas de naturais tem cardinal $\aleph_0$

desafio : escolher ao acaso 5 números distinctos com 5 dígitos
haverá uma forma sistemática de construir um **sexto** número com 5 dígitos **seguramente diferente** dos 5 primeiros?

a resposta é sim e foi o próprio Cantor que mostrou como: **a técnica da diagonal**

1	4	2	7	5
4	6	2	8	1
7	5	1	3	6
5	4	3	7	1
3	3	3	3	3

para formar um número de 5 dígitos **distintos de cada um** dos 5 números aqui apresentados, basta que este difere **guaratidamente** de um dígito de cada um dos outros

basta assim considerar o número que está na diagonal da matriz (assim contempla exactamente um digitos de cada um dos 5 números) e propositadamente alterá-lo, por exemplo somando um (módulo 10)

1	4	2	7	5
4	**6**	2	8	1
7	5	**1**	3	6
5	4	3	**7**	1
3	3	3	3	**3**

$\Longrightarrow$ 16173 $\Longrightarrow$ **27284**

este argumento funciona para todas colecções (numeráveis) que se organizam em **matrizes quadradas**

(os índices das linhas formam segmentos iniciais de inteiros)

quando consideradas infinitas, podemos usar as matrizes resultantes e este princípio ao nosso proveito para **mostrar a não equipotência**

e se considerarmos sequências infinitas de números contáveis?

tomemos o caso das sequências de bits (0s e 1s)

assumindo que são também contáveis (i.e., o conjunto tem cardinal $\aleph_0$),

vamos organiza-las todas numa matriz seguindo uma determinada ordem de contagem que por hipótese existe:

- na primeira linha fica a primeira sequência de bits (por exemplo a que só tem zeros)
- na segunda linha fica a segunda sequência bits na enumeração escolhida
- etc.

a matriz é teórica: contém uma enumeração de todas as sequências possíveis de 0s e 1s na ordem seguida na contagem que garante que é numerável

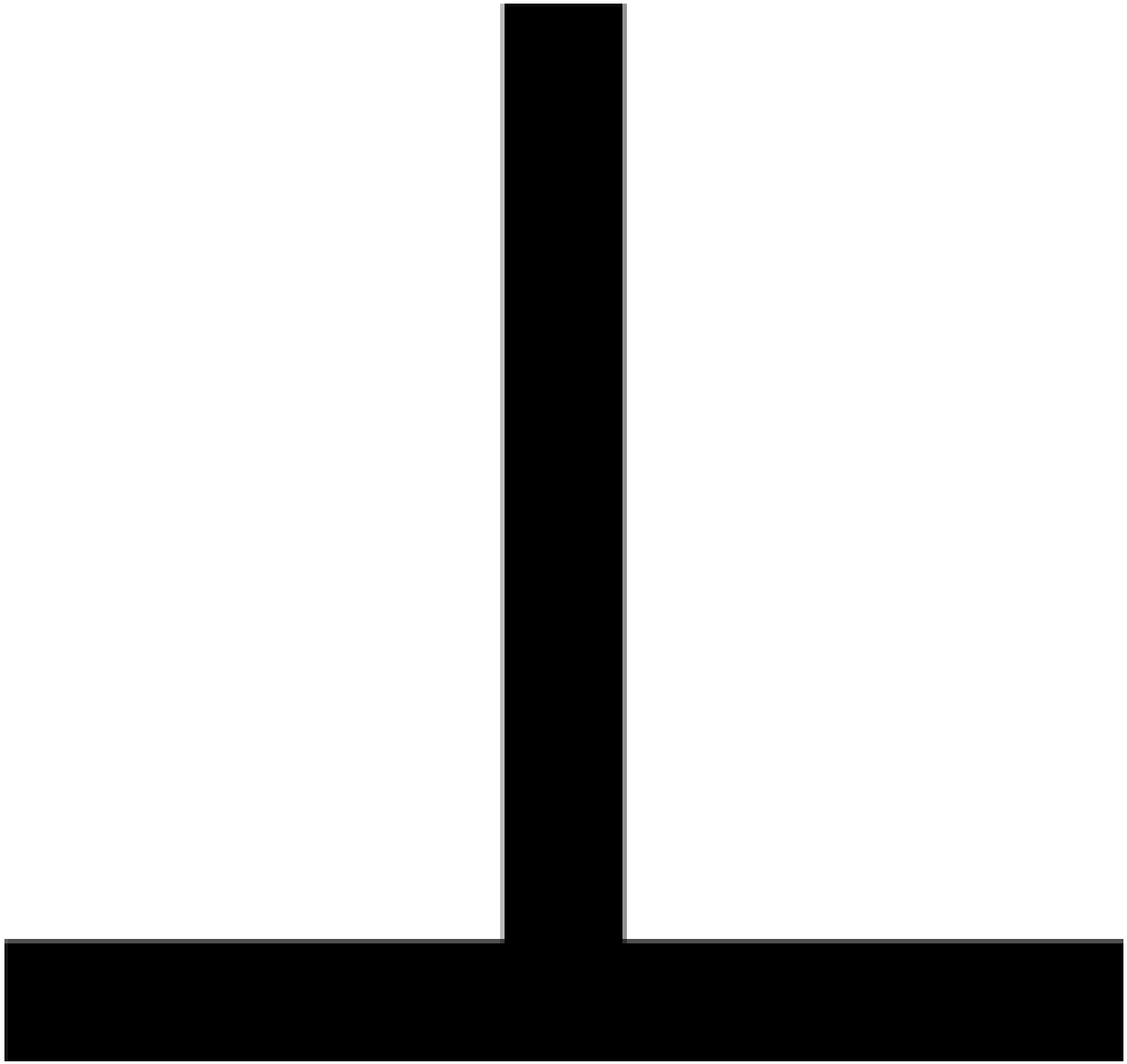
a **diagonal desta matriz** é também uma sequência de bits infinitas então **pertence a matriz**

o **complemento da diagonal** (trocar os 0s por 1s e vice-versa) é também uma sequência de bits infinitas então **pertence a matriz**

mas, como a **técnica da diagonal** nos mostrou, sabemos que **é diferente de todas as sequências** da matriz (porque difere de pelo menos de um bit de todas elas)

assim sendo encontramos uma sequência que **simultaneamente está e não está** na matriz de todas as sequências infinitas de bits

mas... então?... é enumerável ou não...




BAM!

a cada conjunto infinito de sequência de bits, conseguimos juntar mais uma sequência

o conjunto infinito das sequências de bits **não é equipotente** ao conjunto $\mathbb{N}$, logo não é contável e tem um cardinal superior a $\aleph_0$!

com um raciocínio semelhante, considerando as sequências de algarismos que representam a parte decimal de qualquer real maior ou igual a zero e menor que 1,

concluimos que o conjunto $[0,1[$ tem cardinal **superior** a $\aleph_0$.

que valor é este?

note-se que o conjunto de todas as sequências de naturais é o conjunto das partes dos naturais, ou seja, o produto cartesiano contável de naturais.

o seu cardinal, chamemos-lhe $\aleph_1$, é $2^{\aleph_0}$

fazendo agora um conjunto das partes de um conjunto com cardinal $\aleph_1$, obtemos um conjunto com cardinal $\aleph_2$, ou seja, $2^{\aleph_1}$

iterando este processo obtemos uma sequência infinita crescente de números cardinais

há **infinitos infinitos**!

Conclusão. Quer saber mais?

este material apresentado foi realizado com base nas seguintes referências, umas lúdicas, outras mais técnicas, todas relevantes.

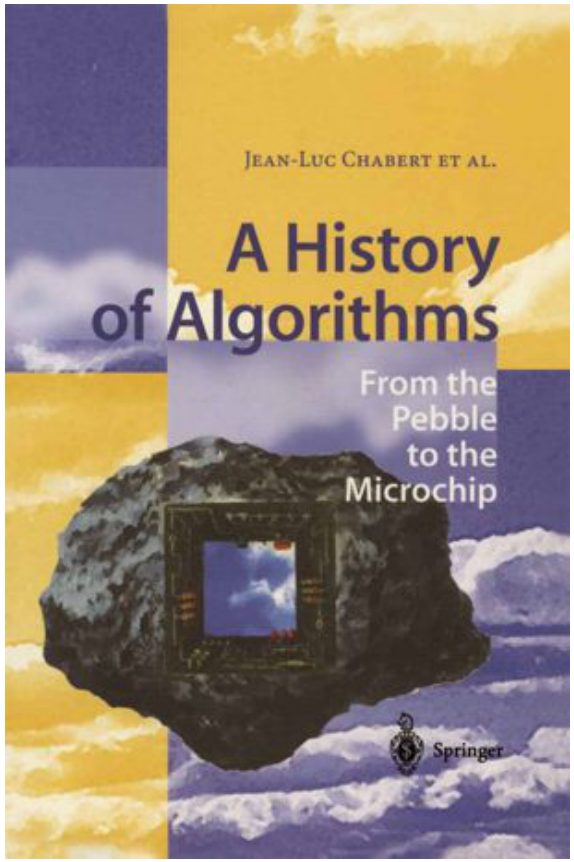
sebenta "Introdução à Teoria da Computação" (url)

esta sebenta tem mais algum detalhe sobre esta introdução à (teoria da) Computação

a introdução às aulas de Lógica Computacional (com a autoria de António Ravara) também contém material aqui apresentado

este material apresentado foi realizado com base nas seguintes referências, umas lúdicas, outras mais técnicas, todas relevantes.

A History of Algorithms
From the Pebble to the Microchip
Editors: Chabert, Jean-Luc (Ed.)
Springer-Verlag, 1999



este material apresentado foi realizado com base nas seguintes referências, umas lúdicas, outras mais técnicas, todas relevantes.

Introduction à l'informatique.

Isabelle Tellier

2001

sebenta disponível online - (URL)

Este material apresentado foi realizado com base nas seguintes referências, umas lúdicas, outras mais técnicas, todas relevantes.

LOGICOMIX
Uma busca épica pela verdade

Christos Papadimitriou, Apostolos Doxiadis

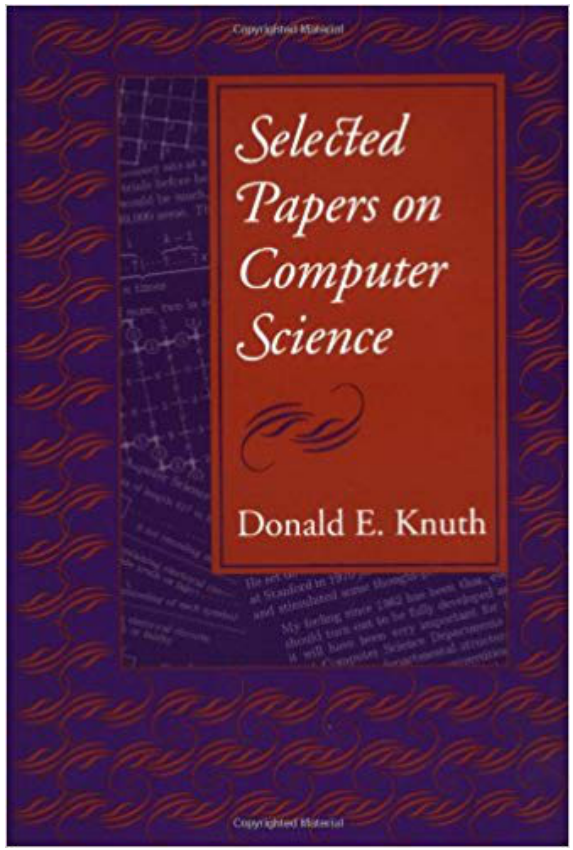
p. 352, Gradiva , 2014
ISBN: 9789896166014

este material apresentado foi realizado com base nas seguintes referências, umas lúdicas, outras mais técnicas, todas relevantes.

Selected papers on computer science,

Donald E. Knuth

Center for the Study of Language and Information,
Fourth Printing, 2014

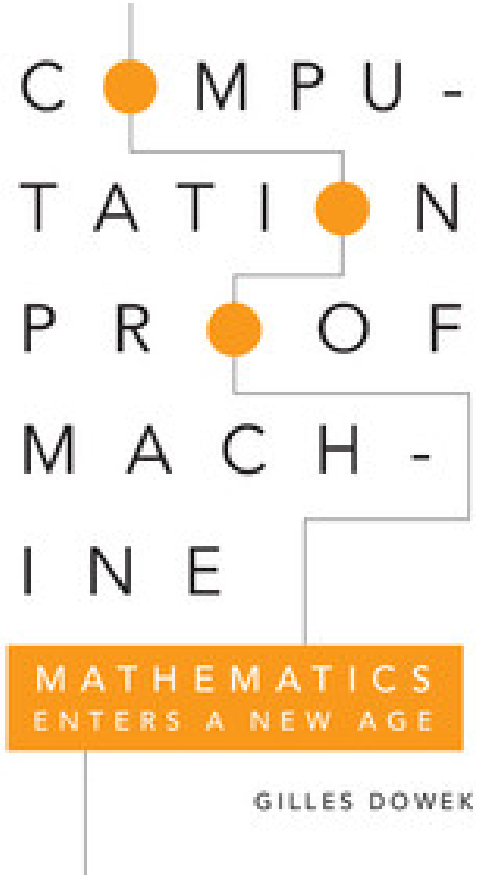


este material apresentado foi realizado com base nas seguintes referências, umas lúdicas, outras mais técnicas, todas relevantes.

Computation, Proof, Machine,

Gilles Dowek

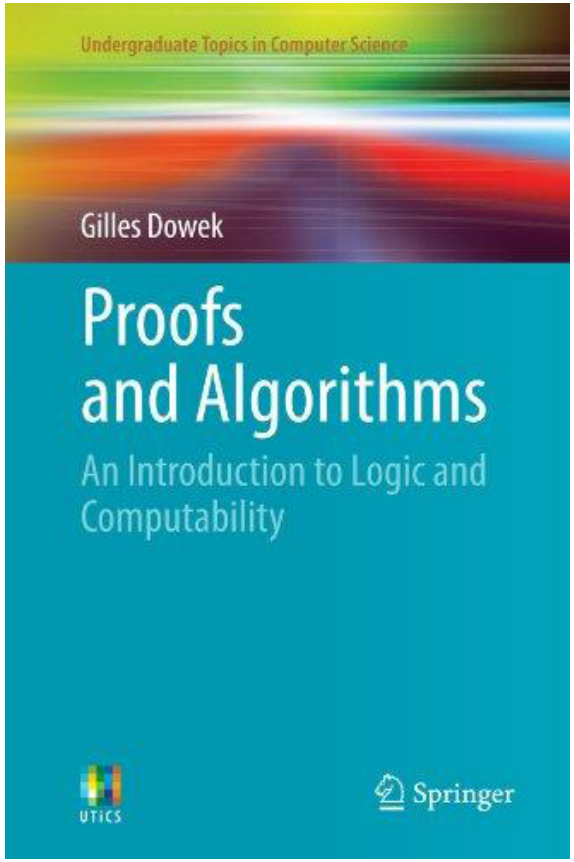
Cambrigde University Press, 2015.



aconselhamos igualmente estas referências complementares

Proofs and Algorithms
An Introduction to Logic and Computability,

Gilles Dowek
Springer-Verlag, 2011.



aconselhamos igualmente estas referências complementares

Turing. Este génio foi decisivo para derrotar Adolf Hitler

História, n.11
Jornal de Notícia

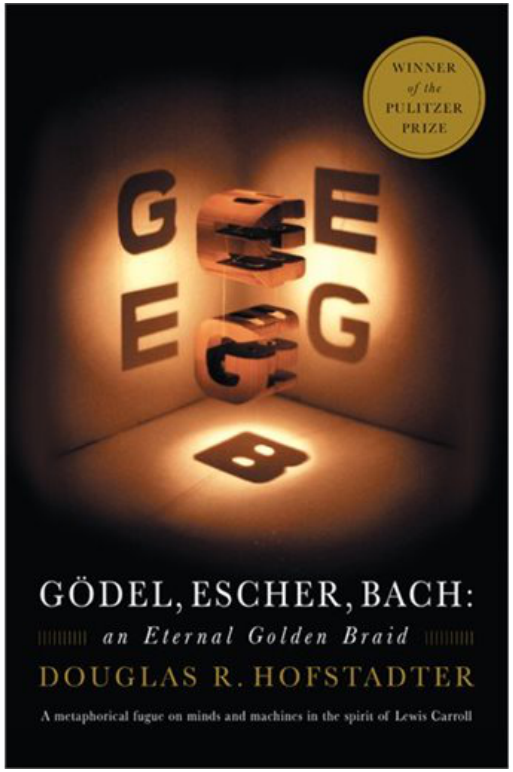
online - (URL)

aconselhamos igualmente estas referências complementares

Gödel, Escher, Bach
An Eternal Golden Braid

D. R. Hofstadter.

Basic Books, New York, 1979.



aconselhamos igualmente estas referências complementares

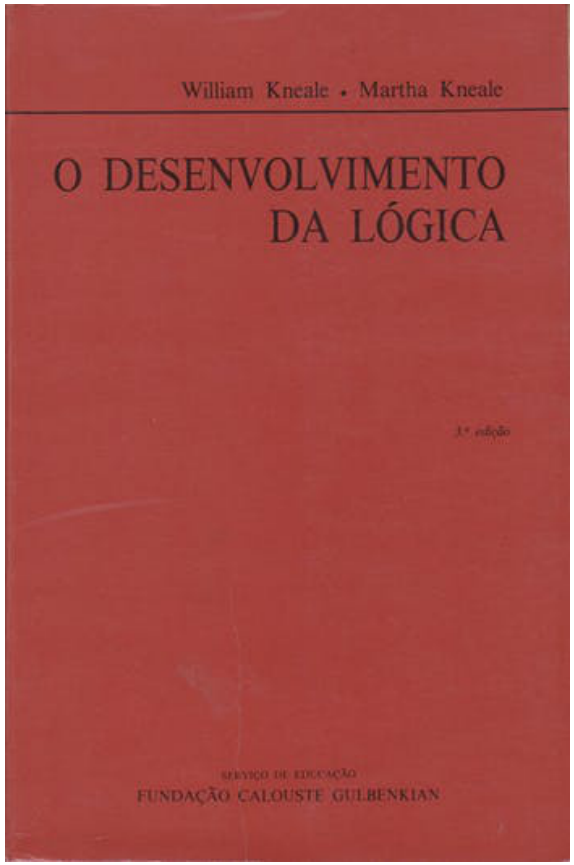
a história da informática (link)

aconselhamos igualmente estas referências complementares

O Desenvolvimento da Lógica

W. Kneale and M. Kneale.

Fundação Calouste Gulbenkian, (1962) - 1991.
Terceira Edição.



jiu-jitsu

(jutsu) técnica, arte ou procedimento
(jiu) para ser macio, flexível, adaptável,

algo ritmo

(ritmo) técnica, arte, contorção ou dança
(algo) para obter algo