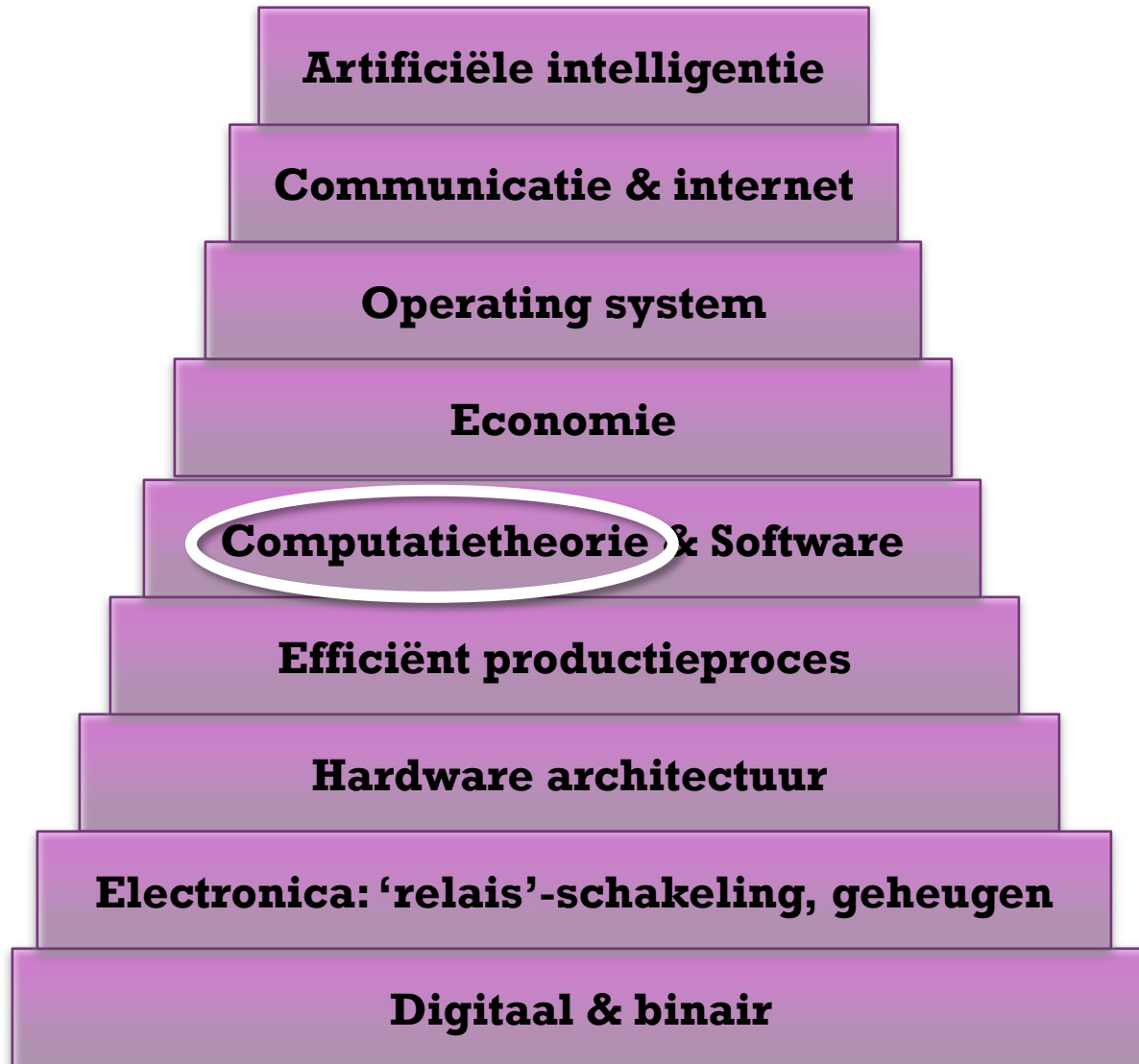
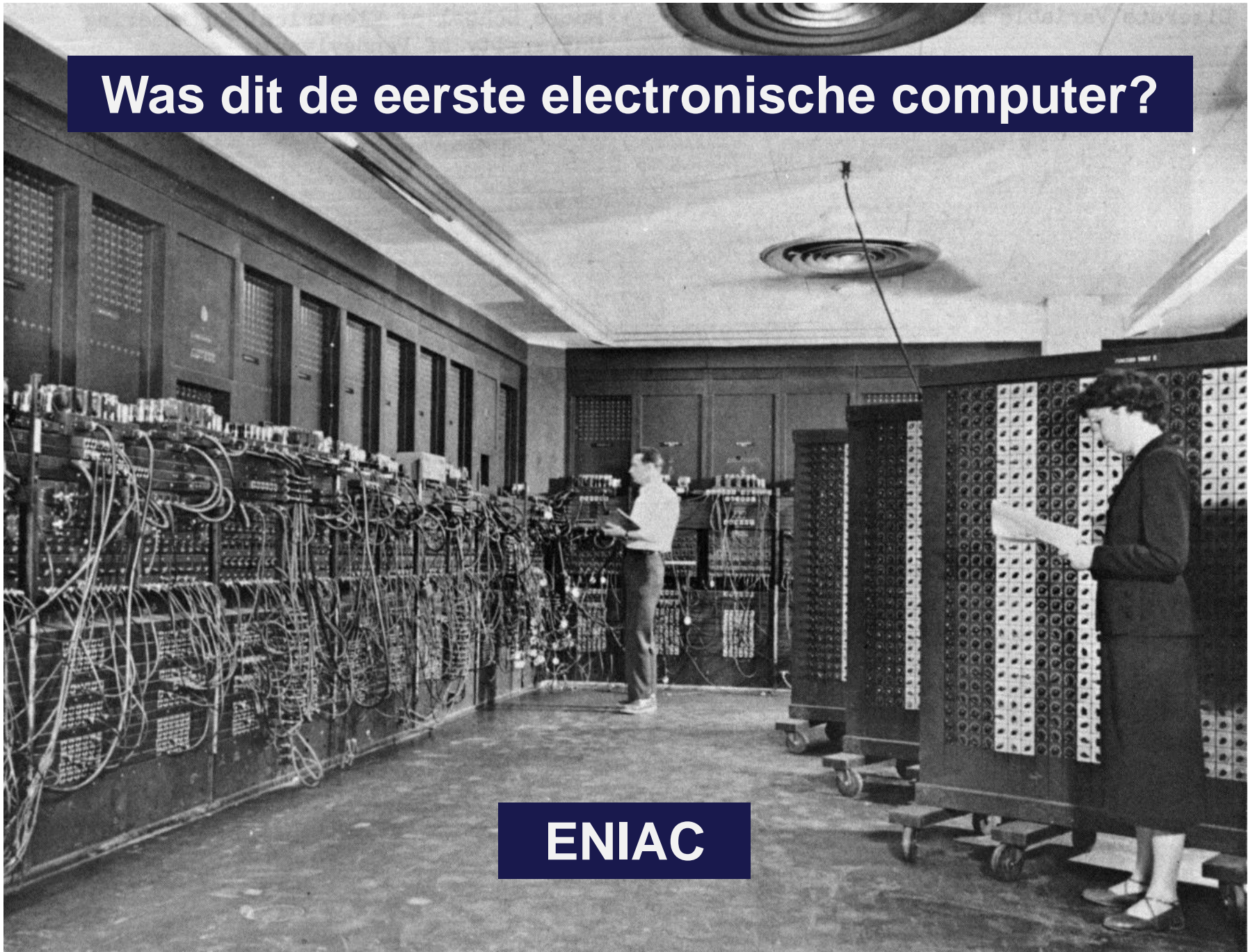


Waarmaken van Leibniz's droom

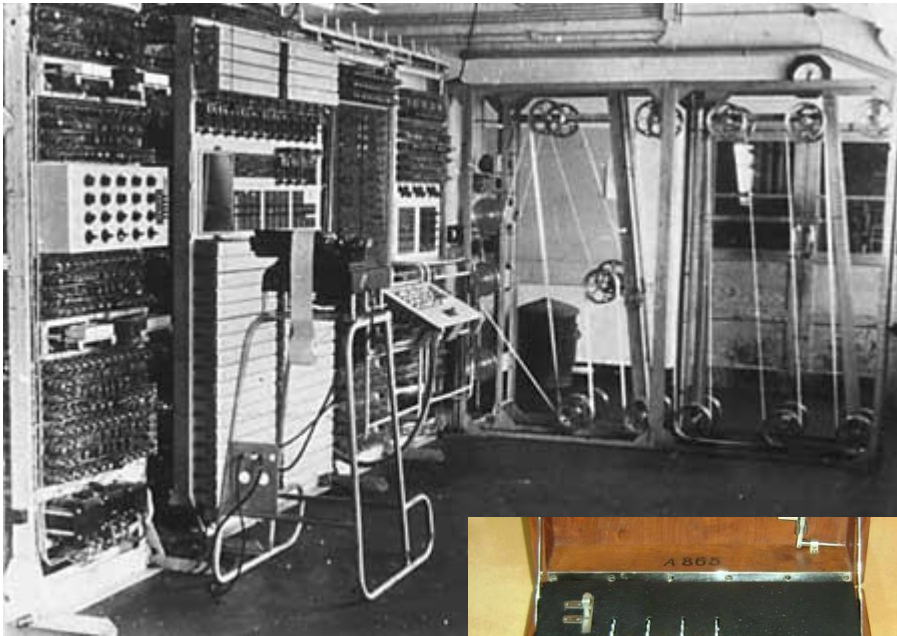


Was dit de eerste electronische computer?



ENIAC

Colossus: the first operational electronic computer



Alan Turing
1912-1954



Leibniz' droom

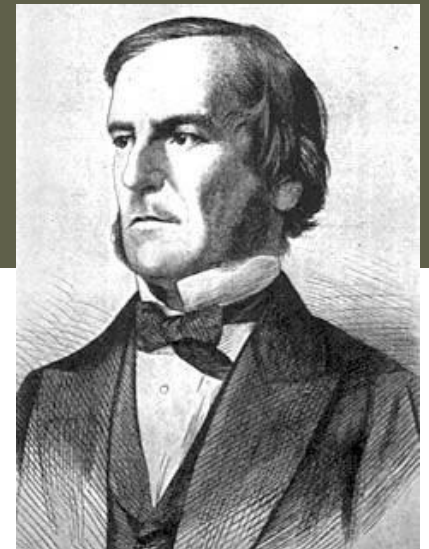


1646 – 1716

◆ De “**Calculus ratiocinator**”

- ◆ Een logisch denkend apparaat
- ◆ Om met de regels van de logica ondubbelzinnig te kunnen vaststellen of een statement waar of vals is
- ◆ Deductief systeem waarin alle regels afgeleid zijn van een kleine verzameling axiomas

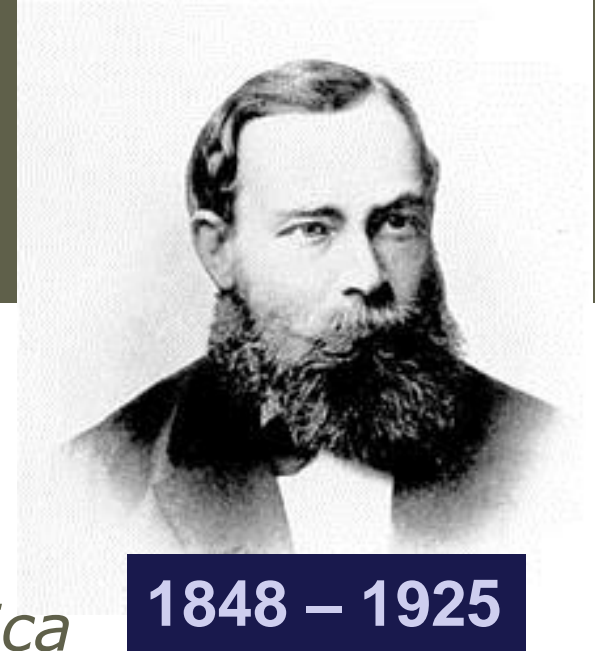
George Boole



1815 – 1864

◆ Logica herleid tot Booleaanse algebra

Gottlob Frege



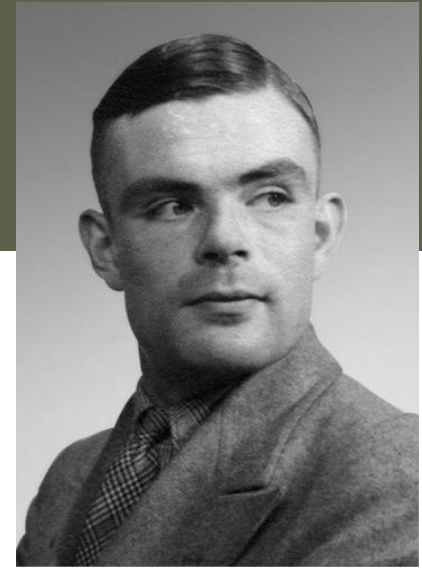
- ◆ Beschrijft hoe wiskundigen en logici denken
- ◆ 1879: *de universele taal van de logica*
- ◆ 1903: brief van Bertrand Russel die hem op een onvolledigheid wijst
 - ✦ Extra-ordinaire verzameling = verzameling dat een element is van zijn eigen
 - Vb: “verzameling van alle dingen die geen spreekw zijn”
 - ✦ *Maar*: de verzameling ***E*** van alle ordinaire verzamelingen
 - En wat is ***E***? Ordinair of extra-ordinair

David Hilbert



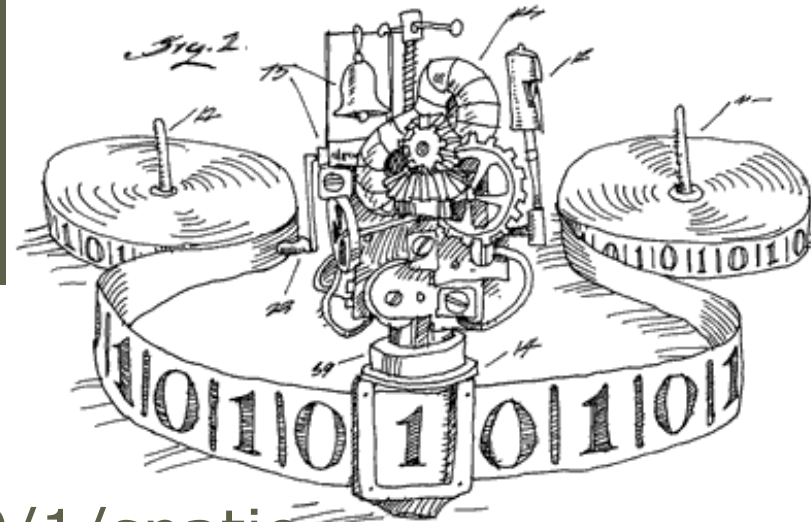
- ◆ “Wir müssen wissen; wir werden wissen”
 - ✦ We moeten weten; we zullen weten
- ◆ Elke logische of mathematische vraag is oplosbaar (en zal opgelost worden)
- ◆ *Nodig*: een expliciete procedure om vanuit premises na te gaan of een conclusie volgt of niet (Hilbert’s **beslissingsprobleem**)

Alan Turing



- ◆ Hilbert's procedure: **algoritme!**
- ◆ Als we een 'mechanische' lijst van regels hebben om de oplossing van een mathematisch probleem op te lossen, zouden wiskundigen geen werk meer hebben...
- ◆ Bestaat er zo'n algoritme?
- ◆ Om tegendeel te bewijzen moest hij een ding maken dat een/elk algoritme kan uitvoeren

De Turing machine



- ◆ **Data:** oneindige lange tape voor het lezen of schrijven van 0/1/spatie
 - ✦ Kan 1 vakje naar links of rechts bewogen worden
- ◆ **Staat (toestand):** een aantal binaire variabelen
- ◆ **Gedragsregels**

Staat	Gelezen bit	Nieuwe staat	Te schrijven bit	Bewegen
0	0	0	0	>
0	1	1	1	<
1	0	1	—	>
1	1	HALT		

Voorbeeld

◆ Applet van studenten 2012

◆ Java applet:

<http://ironphoenix.org/tril/tm/>

Oplossing van Hilbert's beslissingsprobleem?

- ◆ Met de machine nagaan of een logisch statement waar of vals is:
 - ✦ Programma runnen
 - ✦ Als de machine stopt is statement bewezen (waar).
 - ◆ '**Halting problem**' (tegelijkertijd ontdekt door Alonzo Church en Alan Turing)
 - ✦ Gegeven een Turing machine en programma, ga na of dit programma zal stoppen.
 - ✦ Hier bestaat **geen** algoritme voor die dit kan nagaan
- ➔ *Hilbert's beslissingsprobleem is onoplosbaar*
- ➔ *Turing's resultaten wel belangrijk voor informatica*

De universele computer

- ◆ Elke computatie kan er op uitgevoerd worden = Turing Machine
- ◆ Kan alles berekenen wat berekenbaar is
 - ◆ rekenen
 - ◆ ook om logisch te denken...
 - ➔ universeel

Alan Turing



Alan Turing
1912-1954

- ◆ Universele computer
 - ◆ Om te rekenen
 - ◆ Ook om logisch te denken...

- ◆ Basis van de theoretische informatica



◆ Howard Aiken, 1956

"If it should turn out that the basic logics of a machine designed for the numerical solution of differential equations coincide with the logics of a machine intended to make bills for a department store, I would regards this as the most amazing coincidence I have ever encountered"

In 1947:

"slechts zes elektronische digitale computers zullen voldoende zijn om al het rekenwerk van de gehele Verenigde Staten te kunnen doen."

Basisprincipe

- ◆ 1 universele computer kan elke andere computer simuleren
- ◆ Zo is je programma ook maar data die zegt wat de computer moet doen. Voor dat programma heb je geen aparte machine nodig, toch?
- ◆ Nogal logisch, niet?

Computatietheorie: bewijzen dat een programma correct is...

- ◆ Heel omslachtig!
- ◆ Enkel mogelijk voor eenvoudige algoritmes.
- ◆ *Dus*: ad-hoc testen noodzakelijk, zorgvuldig programmeren, goed opdelen, ...