

AI: Kunstmatige Intelligentie

week 1: Introductie en motivatie

Jan Martin Jansen

Wie ben ik?

Jan Martin Jansen

Studie

- wiskunde en natuurkunde UvA
- promotie Informatica Radboud Universiteit

Werk

- Philips Research
- Noordelijke Hogeschool Leeuwarden
- KM Maritieme IT
- Nederlandse Defensie Academie



Opzet Cursus

1. Introductie:
 - a. Motivatie, geschiedenis, klassieke AI en voorbeelden
2. Moderne AI:
 - a. Neurale netwerken
 - b. Toepassingen: beeld- en spraak-herkenning
 - c. Prompt engineering
3. Taalmodellen en Toepassingen
 - a. Chat applicaties
 - b. Hybride Modellen
4. AI en de Toekomst?
 - a. Ethische aspecten en gevaren AI
 - b. Toekomstscenario's

■ 11:28

OpenAI zet voor 400 miljard dollar vijf nieuwe datacenters neer

De miljardenaankondigingen rondom OpenAI volgen elkaar in rap tempo op. Het AI-bedrijf van Sam Altman gaat de komende [vijf jaar vijf nieuwe grote datacenters neerzetten](#) in de Verenigde Staten, voor in totaal 400 miljard dollar. Dat geld wordt opgehoest met hulp van techbedrijf Oracle en het Japanse Softbank, zo maakten de bedrijven bekend.

De aankondiging volgt kort nadat bekend was geworden dat het Amerikaanse chipbedrijf Nvidia [100 miljard dollar gaat investeren](#) in de maker van ChatGPT.

OpenAI  @OpenAI · Follow

More compute in the making.

Announcing 5 new Stargate sites with Oracle and SoftBank, putting us ahead of schedule on the 10-gigawatt commitment we announced in January.



V Home Krant Net binnen Rubrieken

Week 1

Onderwerpen

- motivatie: wat is er aan de hand?
- geschiedenis computers en AI
- symbolische AI
- voorbeelden symbolische AI systemen
 - wiskunde bewijzen
 - vertalen
 - handschriftherkenning
 - spellen: schaken

■ NIEUWS

OpenAI claimt doorbraak met model dat kan 'redeneren'

Kunstmatige intelligentie OpenAI heeft zijn nieuwe AI-model gepresenteerd. Het model (o1 genaamd) kan 'redeneren', schrijft het Amerikaanse techbedrijf.

Uit NRC

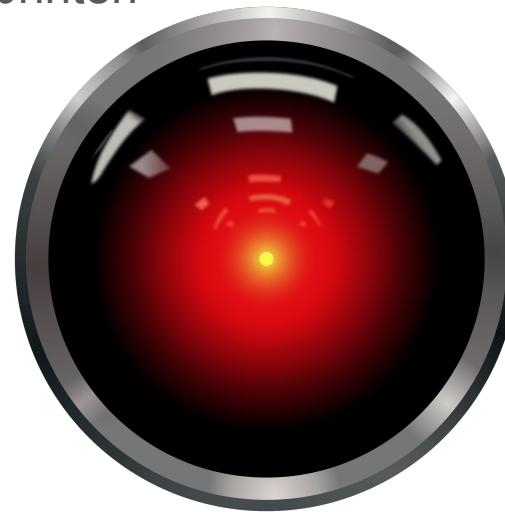
Artificiële Intelligentie



- oude Grieken: Talos de mechanische reus
- robots: Archie de man van staal
- machines die kunnen spreken
- zelf rijdende auto's
- apparaten die gesproken teksten printen
- 2001 A Space Odyssey: Hal
- Star Trek: Computer



Was ooit allemaal science fiction!



Motivatie

Wat is er aan de hand?

- AI is een hype!
- snelle opkomst aantal nieuwe producten
 - **chatGPT**, Dall E, **Gemini**, co-pilot, Perplexity, Claude, etc
 - interactie in gewone taal: vraag en antwoord
- indrukwekkende resultaten
 - generatie van teksten, plaatjes, filmpjes, geluid
 - vertalen van teksten
- hoe kan dat zomaar ineens?
 - evolutie of revolutie?



Wat verwacht men van AI?

Belangrijke toepassingsgebieden voor AI zijn:

- herkennen van beelden
- begrijpen van taal
- automatisch vertalen
- genereren van taal
- besturen van machines (auto's, robots, etc)
- assisteren van mensen bij taken
- overnemen van moeilijke taken
-



Geschiedenis

AI en computers: hoe zit dat nu?

- tot 1940 alleen mechanische rekenmachines
- eerste programmeerbare computer, de Z3, gemaakt door Conrad Zuse
- 1943 de **Colossus** in UK, eerste programmeerbare **digitale** computer (kraken Duitse codes)
- drijvende kracht: problemen uit de tweede wereldoorlog
 - berekeningen atoombom
 - berekenen kogelbanen
 - kraken codes



Door NASA - <https://www.nasa.gov/feature/jpl/when-computers-were-human>, Publiek domein, <https://commons.wikimedia.org/w/index.php?curid=57797050>

Ontwikkeling Computer

- **Colossus** (1943): 2400 vacuümbuizen
- **Eniac** (1945): 18.000 vacuümbuizen, eerste echte computer

Uitvinding transistor 1947

- **TX-0** (1956): 3600 transistoren, eerste experimentele computer met transistoren
- **IBM 7090** (1959): 50.000 transistoren, eerste commerciële computer met transistoren, toepassingen vooral in ruimtevaart en defensie
- **IBM 7030** (1961): 170.000 transistoren, grootste computer met alleen transistoren, vooral voor nucleaire simulaties



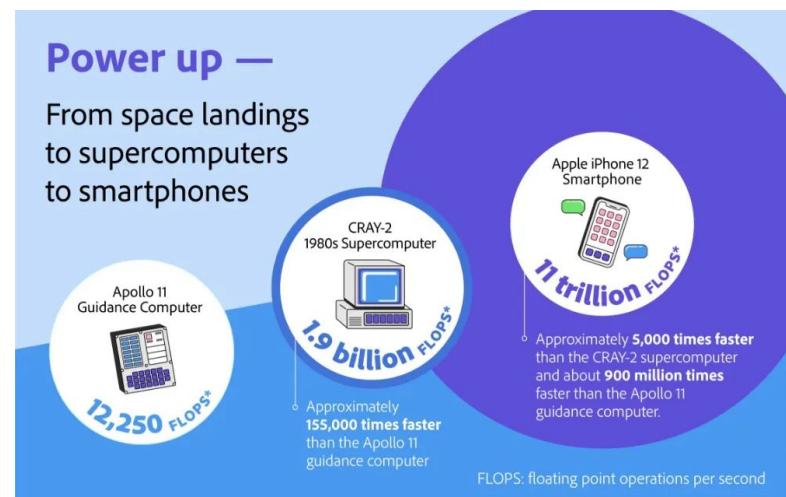
Ontwikkeling Computer

Geïntegreerde Circuits 1958-1959: meerdere transistoren op 1 chip

- eerste computer met IC's: **Apollo Guidance Computer** (1966) met 5000 IC's en totaal 15.000 transistoren, toegepast in ruimtevaart
- **IBM system/360** model 85(1969): eerste commerciële computer met IC's, 50.000 IC's en 200.000 - 500.000 transistoren

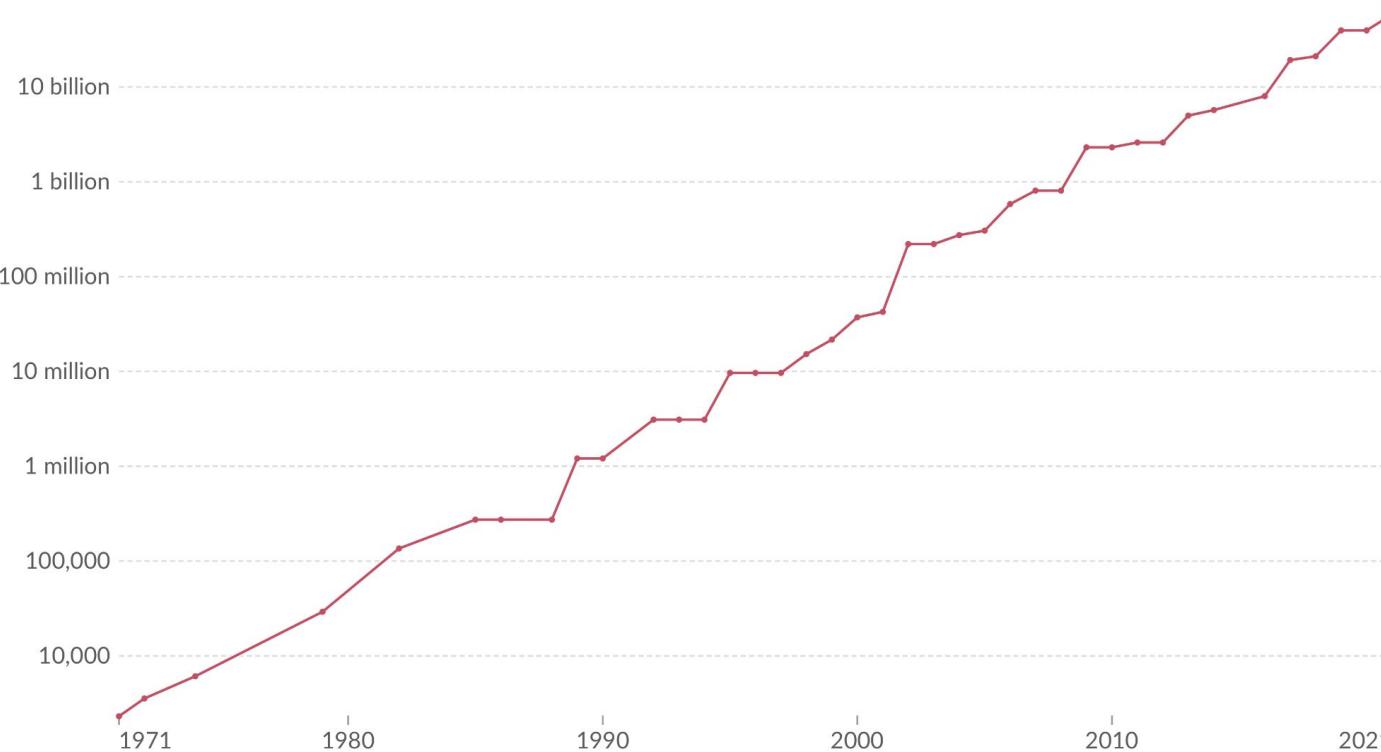
Micro Processor: complete computer op 1 chip

- **Intel 4004** (1971), 2300 transistoren
- **Intel 8088** (1981), 29.000 transistoren, IBM PC
- **Intel 80286** (1982), 120.000 transistoren
- **Intel 80386** (1985), 275.000 transistoren
- **Intel Pentium** (1993), 3.000.000 transistoren
- **Intel Pentium 4** (2002), 400.000.000 transistoren
- **Intel Core I7** (2011), 1.2 miljard transistoren
- **Apple M1** (2021), 16 miljard transistoren
- **NVIDIA H100 Tensor Core GPU** (2022), 80 miljard transistoren



Moore's law: The number of transistors per microprocessor

Moore's law is the observation that the number of transistors in an integrated circuit doubles about every two years, thanks to improvements in production. It was first described by Gordon E. Moore, the co-founder of Intel, in 1965.



Data source: Karl Rupp, Microprocessor Trend Data (2022)

OurWorldInData.org/technological-change | CC BY

- 20 µm – 1968
- 10 µm – 1971
- 6 µm – 1974
- 3 µm – 1977
- 1.5 µm – 1981
- 1 µm – 1984
- 800 nm – 1987
- 600 nm – 1990
- 350 nm – 1993
- 250 nm – 1996
- 180 nm – 1999
- 130 nm – 2001
- 90 nm – 2003
- 65 nm – 2005
- 45 nm – 2007
- 32 nm – 2009
- 28 nm – 2010
- 22 nm – 2012
- 14 nm – 2014
- 10 nm – 2016
- 7 nm – 2018
- 5 nm – 2020
- 3 nm – 2022

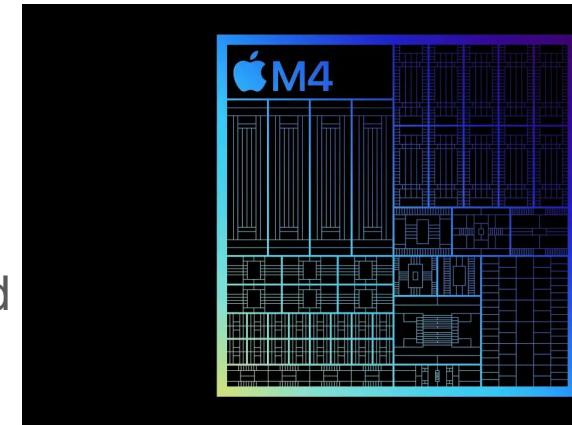
Ontwikkeling Computer

Verwachting

- **1 biljard** (1.000.000.000.000) transistoren rond 2030!

Waarom is dit zo belangrijk?

- moderne processoren zoals de M en A serie van Apple zijn krachtiger dan de duurste supercomputers uit de jaren 80!
- heel veel rekenkracht is de bepalende factor voor moderne AI applicaties!



Wat kan een Computer?

- berekeningen volgens een stappenplan (**algoritme**) uitvoeren
 - mensen voerden stappenplan uit mbv gewone rekenmachines
 - eerste echte computers: plan in hardware vastgelegd (componenten met verbindingen)
 - idee **John von Neumann**: sla het plan op in geheugen (de programmeerbare computer)
- programmeerbare computer
 - voert berekeningen in **stappen** uit
 - **herhaling**: terwijl / tot bepaalde conditie geldt
 - **keuze**: kies afhankelijk van conditie uit alternatieven
 - doe **berekening** en **sla op in geheugen**
 - wat staat er in geheugen computer?
 - alleen maar **0** en **1**!
 - pas op scherm iets leesbaars/zichtbaars

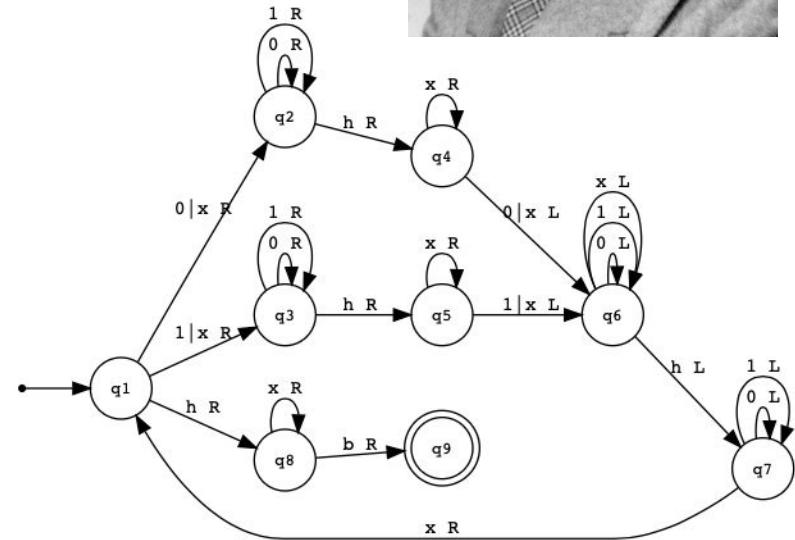
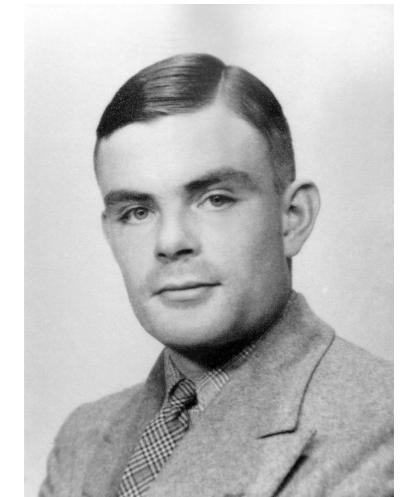


Algorithm

Wat kan een Computer?

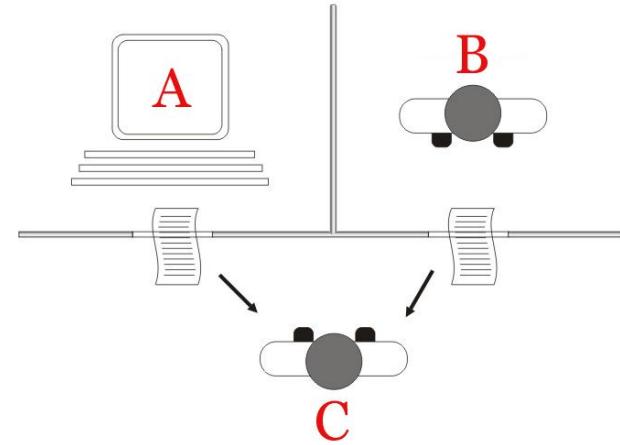
Kan een moderne computer meer dan een computer van vroeger?

- het korte antwoord is **nee**, hij is alleen maar veel en veel sneller!
- Alan Turing bedacht in 1936 het concept van de **Turing Machine**
- dit is een **universeel** computer model
 - een computer kan niet krachtiger zijn dan een Turing Machine
 - dit gaat alleen over het soort berekeningen dat je er mee kunt doen en niet de snelheid!
- het lange antwoord is **ja**, als je een miljard jaar moet wachten op een antwoord en het alternatief is enkele seconden!



Geschiedenis AI in notendop

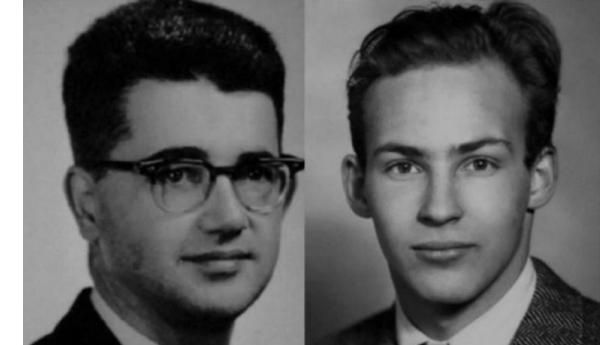
- al tijdens de tweede wereldoorlog probeerde wetenschappers **schaken** te programmeren
- in 1950 schreef **Alan Turing** het artikel "Computing machinery and Intelligence" waarin hij ook de **Turing Test** beschreef:
 - de vraag is of een machine kan **denken**, daartoe:
 - bevraagt persoon C computer A en persoon B via geschreven **berichten** en moet erachter komen wie computer en wie persoon is
 - de test is geslaagd als C het onderscheid niet kan maken!
- Dartmouth workshop (1956)
 - introductie term: Artificial Intelligence



Dartmouth Uitdaging

Uitgangspunt!

- elk **aspect van leren** of een ander kenmerk van **intelligentie** kan in principe zo nauwkeurig worden beschreven dat een **machine** het kan **simuleren**
- probeer te ontdekken hoe machines:
 - taal kunnen gebruiken
 - abstracties en concepten kunnen vormen
 - problemen kunnen oplossen die nu alleen voor mensen zijn weggelegd
 - zichzelf kunnen verbeteren
- aan ambities dus geen gebrek!
 - men verwachtte de geformuleerde doelen **binnen enkele tientallen jaren** te kunnen behalen



John McCarthy, Marvin Minsky
Vaders van AI

Verwachtingen in detail

Wat waren de verwachtingen?

- Machine kan **elke intellectuele taak** die een mens kan uitvoeren overnemen:
 - bewijzen van **wiskundige** stellingen
 - schrijven van **muziek**
 - **wetenschappelijke** ontdekkingen
- Zelflerende machines
 - machines zichzelf **verbeteren**
 - **nieuwe kennis** opdoen zonder voortdurende menselijke tussenkomst.
- Kunstmatige intelligentie binnen een generatie gecreëerd
 - ontwikkeling van machine met echte intelligentie slechts een **kwestie van tijd**

Marvin Minsky in Life 1970

“from three to eight years we will have a machine with the general intelligence of an average human being.”



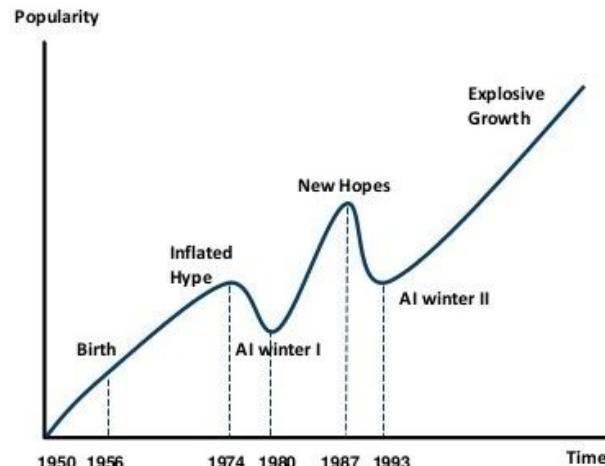
By The original uploader was Sethwoodworth at English Wikipedia. - Transferred from en.wikipedia to Commons by Mardetanha using CommonsHelper., CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=7292026>

Geschiedenis AI in notendop

De verwachting kwam niet uit!

- leidde tot de **AI winter**
- onderzoek naar AI op laag pitje tot midden jaren 80
- daarna in eerste instantie weer bescheiden **opleving**
- betrof in eerste instantie vnl **symbolische AI**

AI HAS A LONG HISTORY OF BEING "THE NEXT BIG THING"...



Timeline of AI Development

- 1950s-1960s: First AI boom - the age of reasoning, prototype AI developed
- 1970s: AI winter I
- 1980s-1990s: Second AI boom: the age of Knowledge representation (appearance of expert systems capable of reproducing human decision-making)
- 1990s: AI winter II
- 1997: Deep Blue beats Gary Kasparov
- 2006: University of Toronto develops Deep Learning
- 2011: IBM's Watson won Jeopardy
- 2016: Go software based on Deep Learning beats world's champions

Symbolische AI

Klassieke benadering van kunstmatige intelligentie

Wat is symbolische AI?

benadering kunstmatige intelligentie door modelleren menselijke kennis en redeneren dmv symbolen en regels.

Kernprincipes:

- **Kennisrepresentatie:** kennis expliciet in formele taal (logica of productie regels)
- **Redeneren:** toepassen regels op symbolen voor nieuwe conclusies
- **Probleemoplossing:** oplossen door het zoeken naar reeks regels van begintoestand naar gewenste eindtoestand

Symbolische AI

Voorbeelden

- **Expertsystemen:** nabootsen menselijke expert op een bepaald gebied
 - **medische** diagnose
 - **financiële** planning
 - configuratie computersystemen
- **Theorembewijzers:** wiskundige stellingen bewijzen door toepassen logische regels
- **Natuurlijke taalverwerking:** gebruik symbolische methoden om syntaxis en semantiek van taal te analyseren
- **Planning en scheduling:** complexe taken opsplitsen in kleinere subtaken en optimale volgorde bepalen.

Symbolische AI

Voordelen van symbolische AI:

- **Transparantie:** redenering makkelijk te volgen en te begrijpen
- **Expliciete kennisrepresentatie:** Kennis expliciet vastgelegd, waardoor systeem makkelijk aan te passen en uit te breiden

Beperkingen van symbolische AI:

- **Kennisverwerving:** tijdrovend en kostbaar om kennis te formaliseren
- **Schaalbaarheid:** moeite met grote en complexe problemen
- **Common sense:** moeite met modelleren van common sense kennis (vaak impliciet en contextueel)

Expertsysteem: Autodiagnose

Kennisbasis:

- **Regels:**
 - Als de auto niet start en er geen geluid hoorbaar is, is waarschijnlijk de accu leeg.
 - Als de motor stottert bij accelereren en het motorlampje brandt, kan het een probleem zijn met de bougies of de brandstofpomp.
 - Als de auto naar één kant trekt en er een slijpend geluid hoorbaar is bij het remmen, zijn de remblokken waarschijnlijk versleten.

Expertsysteem: Autodiagnose

Inferentiemotor: gebruiker vragen gesteld over symptomen auto

Voorbeeld interactie:

- **Systeem:** Start de auto niet?
- **Jij:** Nee
- **Systeem:** Hoort u een geluid wanneer u de sleutel omdraait?
- **Jij:** Nee
- **Systeem:** Op basis van uw antwoorden is de accu waarschijnlijk leeg.

Klassieke Case Studies

We behandelen een aantal case studies met AI technieken van voor 2000

- gebruik van AI in wiskunde
- automatisch vertalen
- klassieke beeldherkenning
- schaken

AI in Wiskunde

Beroemd voorbeeld: het **vierkleurenprobleem**

- Iedere landkaart kan ingekleurd worden met slechts **4 kleuren**, zonder dat aangrenzende landen dezelfde kleur hebben
- open probleem tot 1976
- Kenneth Appel en Wolfgang Haken:
 - reductie tot eindig aantal verschillende gevallen
 - computer gebruikt om deze te controleren
 - 1200 uur rekentijd!
- Bewijs kon pas in 2005 volledig geverifieerd worden mbv het bewijssysteem **Coq**



Coq: assistent voor Wiskunde

Wiskunde bewijzen zijn vaak heel complex en moeilijk te controleren

Coq:

- **formalisme** (taal) om **wiskundige bewijzen** in op te schrijven
- in ontwikkeling sinds jaren 80
- kan controleren of de **stappen** in een bewijs **correct** zijn uitgevoerd
- kan kleine stukken uit bewijs **automatisch** doen
- wordt voortdurend uitgebreid

Coq voorbeeld: $n + (m+1) == (n + m) + 1$

Coq

```
Require Import PeanoNat.
```

```
Theorem plus_n_Sm : forall n m : nat, n + S m = S (n + m).
```

```
Proof.
```

```
  intros n m.
```

```
  induction n as [| n' IHn' ].
```

```
    - simpl. reflexivity.
```

```
    - simpl. rewrite IHn'. reflexivity.
```

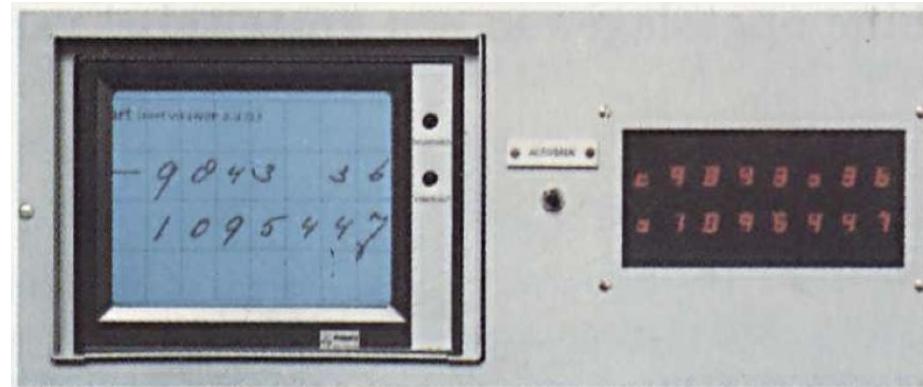
```
Qed.  □
```

Handschrift herkennen

Dr. Neher lab van PTT werkte hier al vanaf 1968 aan!

- toepassingen:
 - gironummers en bedragen herkennen op girokaarten: alleen cijfers en - en X
 - later: postcodes
- 1975: eerste prototype kartonnen girokaarten voor controle mens
 - 5 formulieren per seconde
- 1982: ook voor 'slappe' formulieren
- uitdagend probleem: hoe pak je dat aan?
- eerste computers nog veel te traag
 - eigen hardware
 - koppelen meerdere computer

Handschrift herkennen

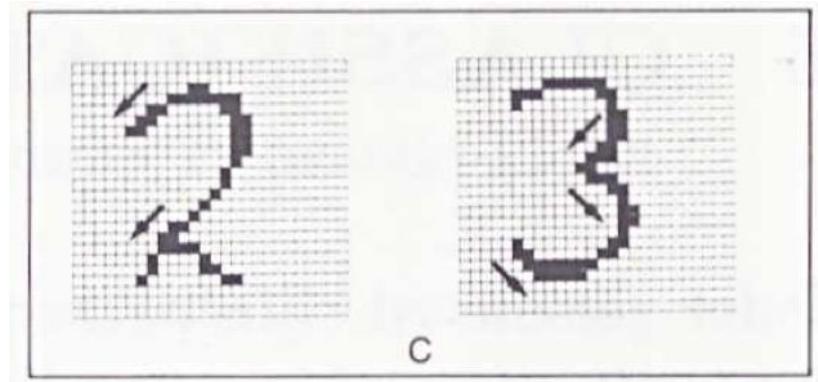


<https://publications.tno.nl/publication/19806495/7Pmp09/essink-1987-computer.pdf>

Handschrift herkennen

Werkwijze, gebaseerd op **kenmerken**:

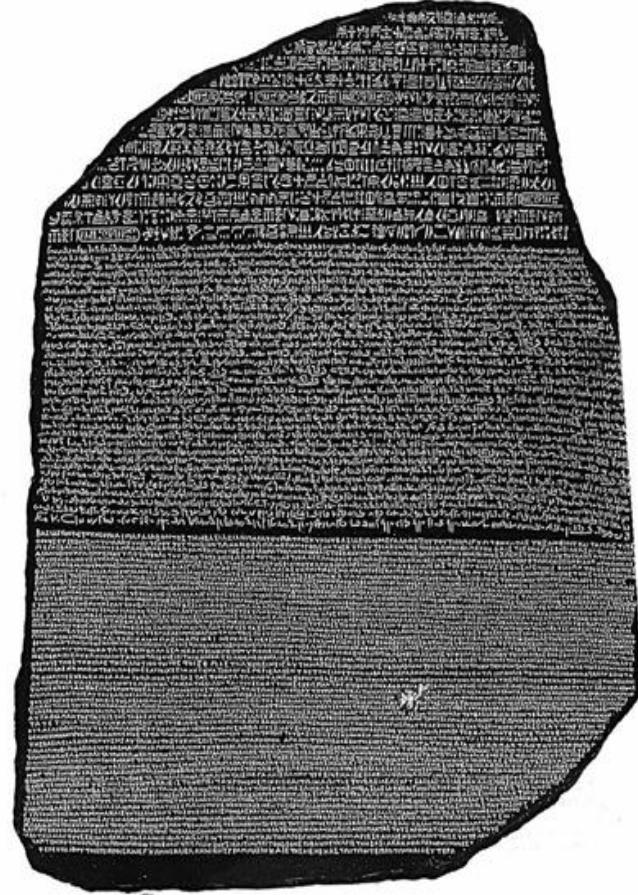
- eindpunten: onder, boven, aantal
- hol/bol: links, recht, onder, boven
- eilanden: rondjes
- sprongen
- combineren van kenmerken
 - totaal 5280 aspecten die bekeken worden
 - tabel voor keuze meest waarschijnlijke letter
 - 98,5% nauwkeurig
- combinatie mens - machine



Automatisch Vertalen

Philips Rosetta Project (jaren 80 en 90)

- ongeveer 20 onderzoekers: ICT-ers en Linguïsten
- multilinguaal: Nederlands, Engels en Spaans
- gebruik grote woordenboeken
- ontleden met grammatica's
 - proberen **betekenis** zin taalonafhankelijk te representeren
- resultaten:
 - redelijke vertaling **losse zinnen**
 - probleem met **contextrijke** teksten



Automatisch Vertalen

Voorbeelden problemen

- ambiguïteit:
 - de man lichtte de bank op (3 betekenissen!)
 - zij zagen het meisje met de kijker (3 betekenissen)
- taal kan erg onregelmatig zijn:
 - veel uitzondering (bv kofschip)
 - en uitzonderingen op uitzonderingen (uitzonderingen kofschip)

Spelletjes

Spel met 2 deelnemers (A en B) die om beurten iets mogen doen

- schaken, dammen, boter-kaas-eieren, go, etc
- zgn **nulsom**-spelen (winst één is verlies ander)

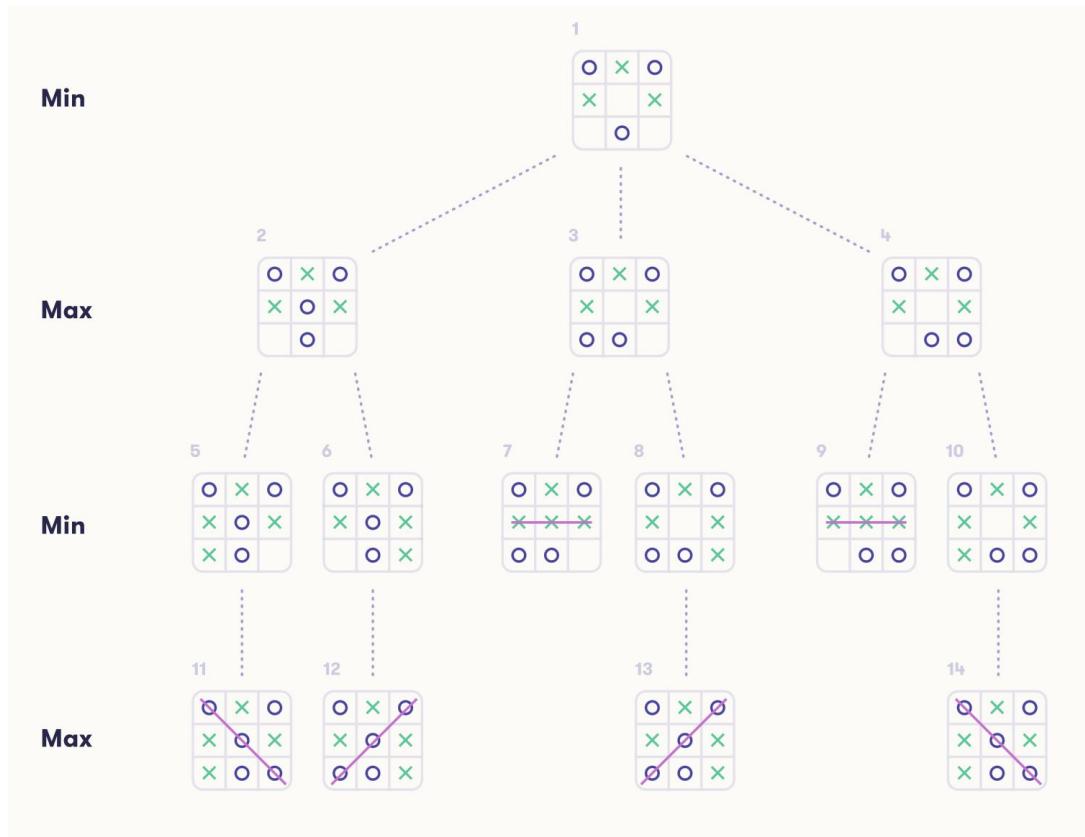


Er bestaat een algoritme voor deze spelen: **Minimax** (of variant alfa-beta)

- iedere positie op het bord krijgt een **waardering** vanuit optiek beide spelers
- er geldt waardering A = - waardering B (nulsom!)
- als speler A aan zet
 - bekijk alle mogelijke zetten die nu gedaan kunnen worden:
 - voor iedere zet bekijk alle mogelijke vervolgzetten van tegenstander
 - kies hieruit steeds zet met hoogste waardering voor B
 - kies nu zelf de zet met de kleinste waardering voor B
- **Minimax** was al bedacht voor er computers waren

Voorbeeld: Spelboom Boter-Kaas-Eieren

O is aan zet



Spelletjes

Minimax levert in theorie altijd een optimale strategie (indien mogelijk)

- genereer complete **spelboom** tot alle mogelijke **eindstanden**
- bepaal van beneden naar boven alle waarderingen (met minimax)
- alle knopen hebben nu een waardering
 - dit is gedaan voor bv checkers (dammen op 8 x 8 bord) in 2007
- probleem complexere spelen: praktisch **onmogelijk** hele boom te berekenen
- oplossing: bepaal waardering met **evaluatie functie** tussengelegen standen
 - gebruik deze van bv 5 -10 zetten vooruit en gebruik daarna minimax
 - de crux zit nu in het maken van een goede evaluatie functie!
 - dit is hoe schaken, dammen, go, etc met computers gespeeld worden (werden)

Schaken

Gebruik minimax algoritme in combinatie met slimme evaluatie functie

- jaren 40,50: correcte zetten met minimale evaluatie
 - wel al uitputtende minimax voor sommige **eindspelen**
 - computer nog niet krachtig genoeg voor beter
- jaren 60: eerste serieuze schaakprogramma's
 - hooguit paar zetten vooruit, simpele evaluatiefunctie
- jaren 70: krachtiger programma's
 - Chess 4.0, MacHack, Belle
 - steeds verder vooruit kijken met betere evaluatie functie
 - eerste **amateur spelers** werden verslagen door computers
- jaren 80: zeer krachtige programma's
 - amateurs vaak geen partij meer
 - 1989: **Hitech** verslaat Mikhail Tal (grootmeester)
- jaren 90: schaakprogramma's worden onverslaanbaar
 - 1997: **Deepblue** verslaat **Gary Kasparov** in een match



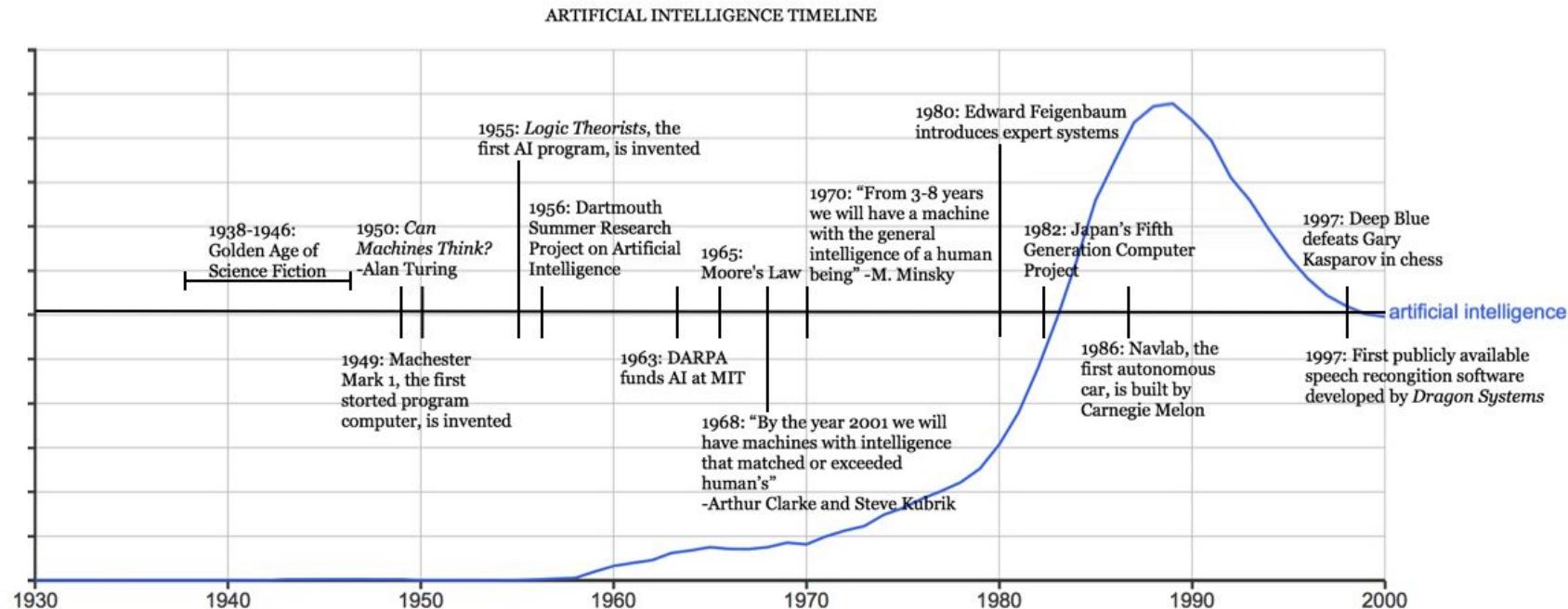
Deepblue

- ontwikkeld door **IBM**
- **200 miljoen** zetten per seconde
- kon tot **40** zetten vooruitkijken (mensen hooguit 6-7)

Bedenk:

- evaluatie functies werden door **mensen** ontwikkeld
- om deze te maken is diepe **kennis** van schaken nodig
- vaak werden ook **databases** met oude partijen gebruikt
- de kracht zit in de **combinatie** van de **evaluatiefunctie** en het ver **vooruit** kunnen kijken!

Samenvatting: Tijdlijn tot 2000



Voorbeelden Interactieve AI chat systemen

OpenAI: chatGPT: <https://chatgpt.com/>

Microsoft: Copilot: <https://copilot.microsoft.com/>

Google: Gemini: <https://gemini.google.com/>

Google: NotebookLM (om documenten te analyseren): <https://notebooklm.google.com/>

Alternatieven:

Anthropic: Claude: <https://claude.ai/>

Perplexity: <https://www.perplexity.ai/>

Veel applicaties zijn ook als App voor een tablet te verkrijgen