

(Kunstmatige) Intelligentie

meer hierover op maandag

(misplaatste?) analogie

- mannetje in de Little Man Computer

vs

- mannetje/vrouwtje in mijn hoofd, oftewel “ik”, dat de boel bestuurt

kunnen machines denken?

- is er een wezenlijk verschil tussen wat wij mensen kunnen en wat machines kunnen ?
- kunnen machines intelligent zijn ?
- hoe zou je dit kunnen testen/meten ?

kunnen machines *schaakspelen*?

- Kasparov is er nog steeds van overtuigd dat Deep Blue hem versloeg dankzij menselijke hulp
- Misschien is voetballen wel een veel betere test voor intelligentie dan schaken?

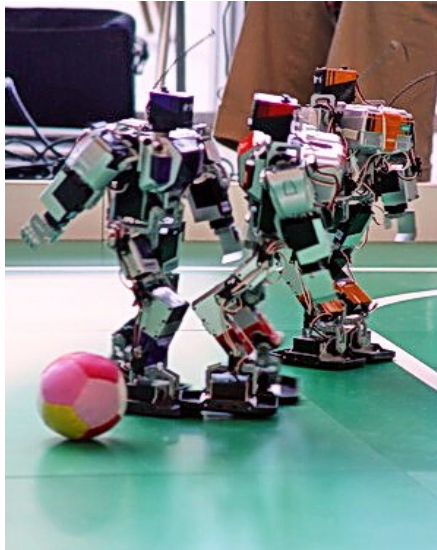
schaakcomputer Deep Blue

- Speciale hardware & software van IBM
 - 200 miljoen zetten per seconde
- 1996: Kasparov wint van Deep Blue (1-2-3)
- 1997: Deep Blue wint van Kasparove (2-3-1)
 - Kasparov denkt dat Deep Blue geholpen is door mensen op grond van “menselijke zetten”



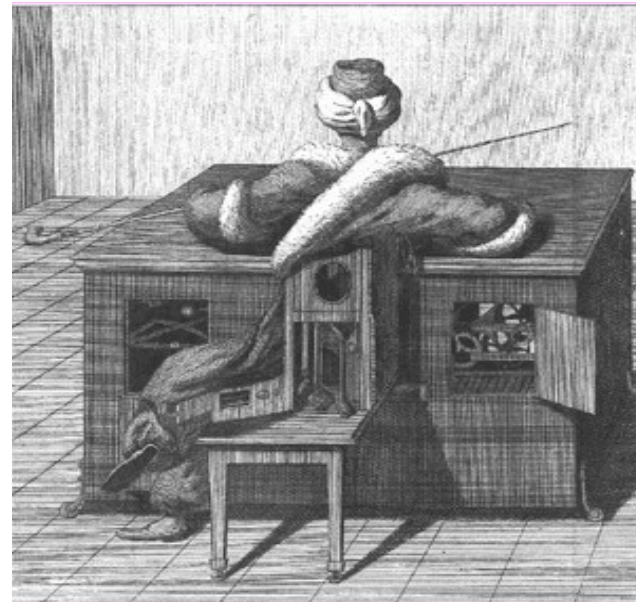
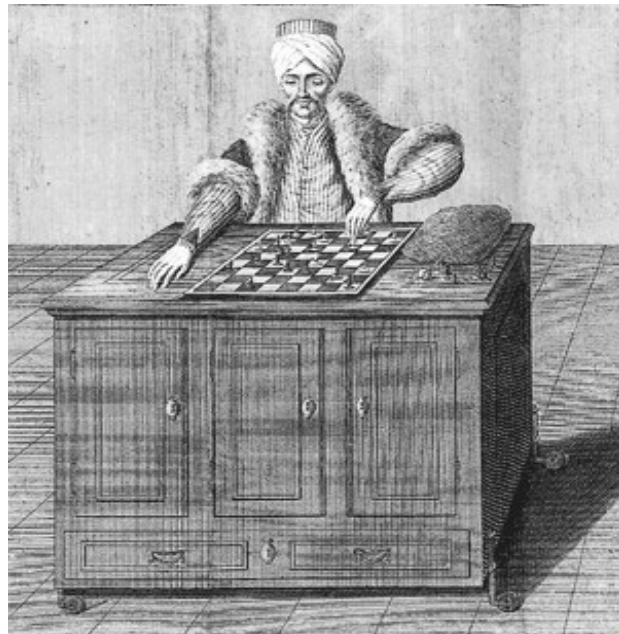
Voetballen of schaken?

- Welke sport vergt meer intelligentie?
- In welke sport zijn computers beter?



Deep Blue anno 1770: de Turk

Mechanische schaakmachine van
Wolfgang von Kempelen



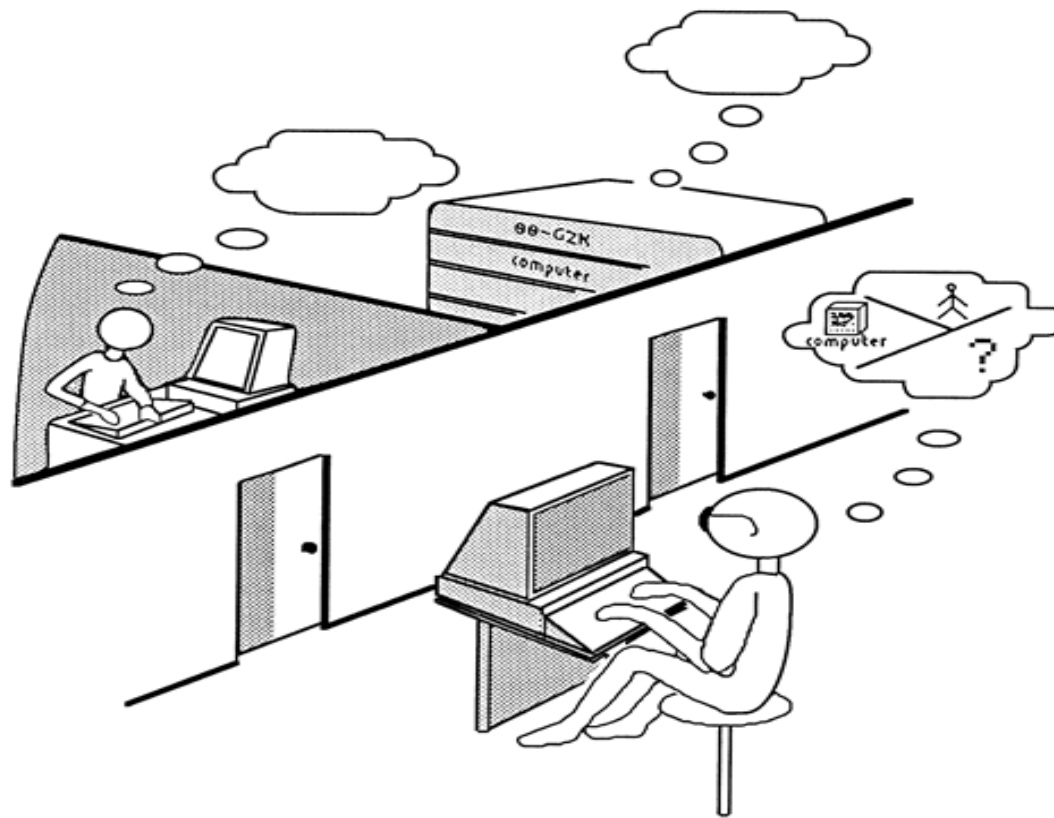
Veel mensen geloofden in 1770 dat zo'n goede
mechanische schaakcomputer mogelijk was!

Turing test

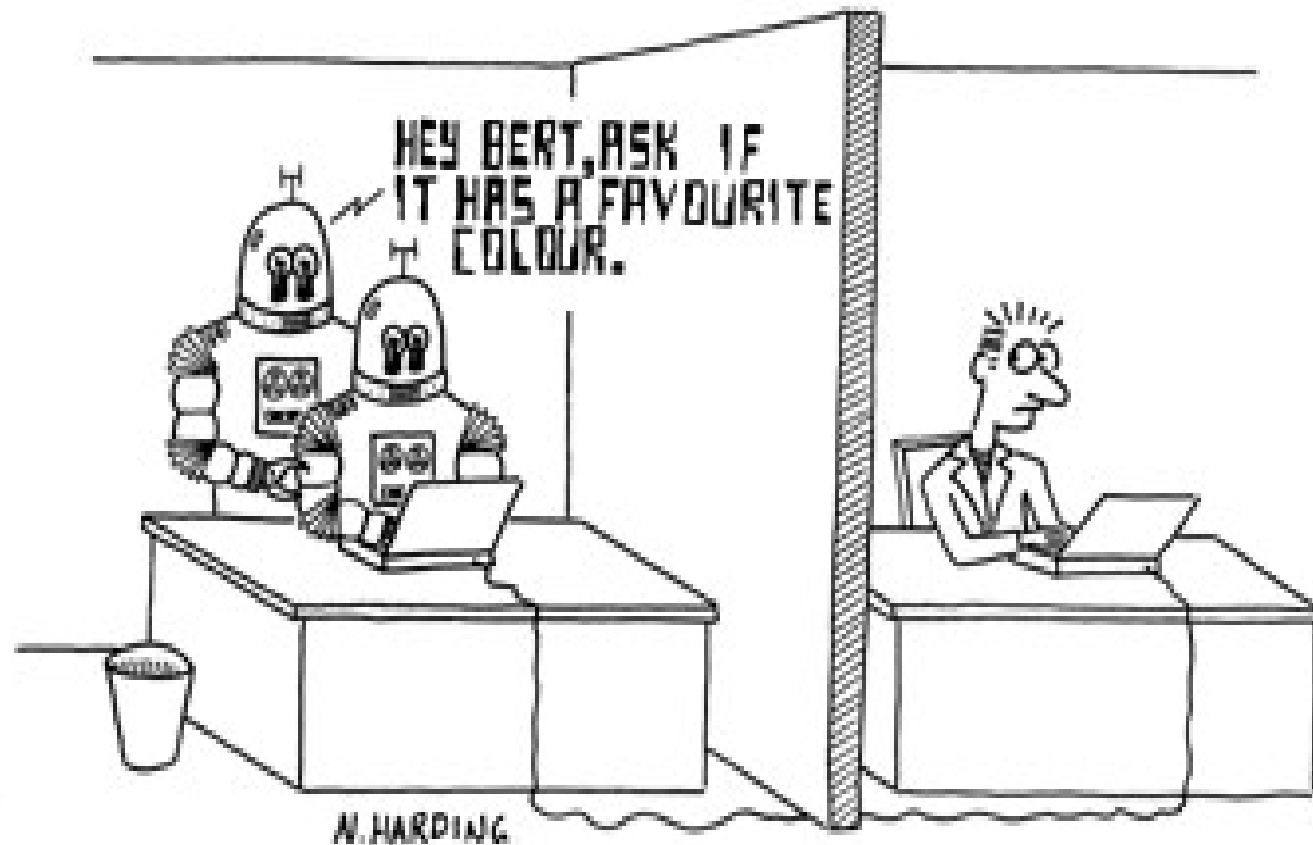
- een machine is intelligent als deze niet te onderscheiden is van een mens in een geschreven vraag & antwoord sessie

[Turing, 1936]

Turing test



Turing test



Turing test in dagelijkse praktijk nu!

Turing test in dagelijkse praktijk nu!

Security Check

Enter **both** words below, separated by a **space**.
Can't read this? [Try another.](#)
[Try an audio captcha](#)



Text in the box:

[◀ Back](#)

Turing test in dagelijkse praktijk nu!

Security Check
Enter **both** words below, separated by a **space**.
[Can't read this? Try another.](#)
[Try an audio captcha](#)



Text in the box:

[◀ Back](#) [Sign Up](#)

- CAPTCHA
- Completely Automated Public Turing-test to tell Computers and Humans and Apart
- omgekeerde Turing test: laat mens bewijzen aan een computer dat-ie mens is

Prediction: The CAPTCHA spam arms race will lead to the creation of A.I.

Are you human? Please answer this question:

~~DESCRIBE~~ what *LOVE*
feels like ~~IN 200~~ words.

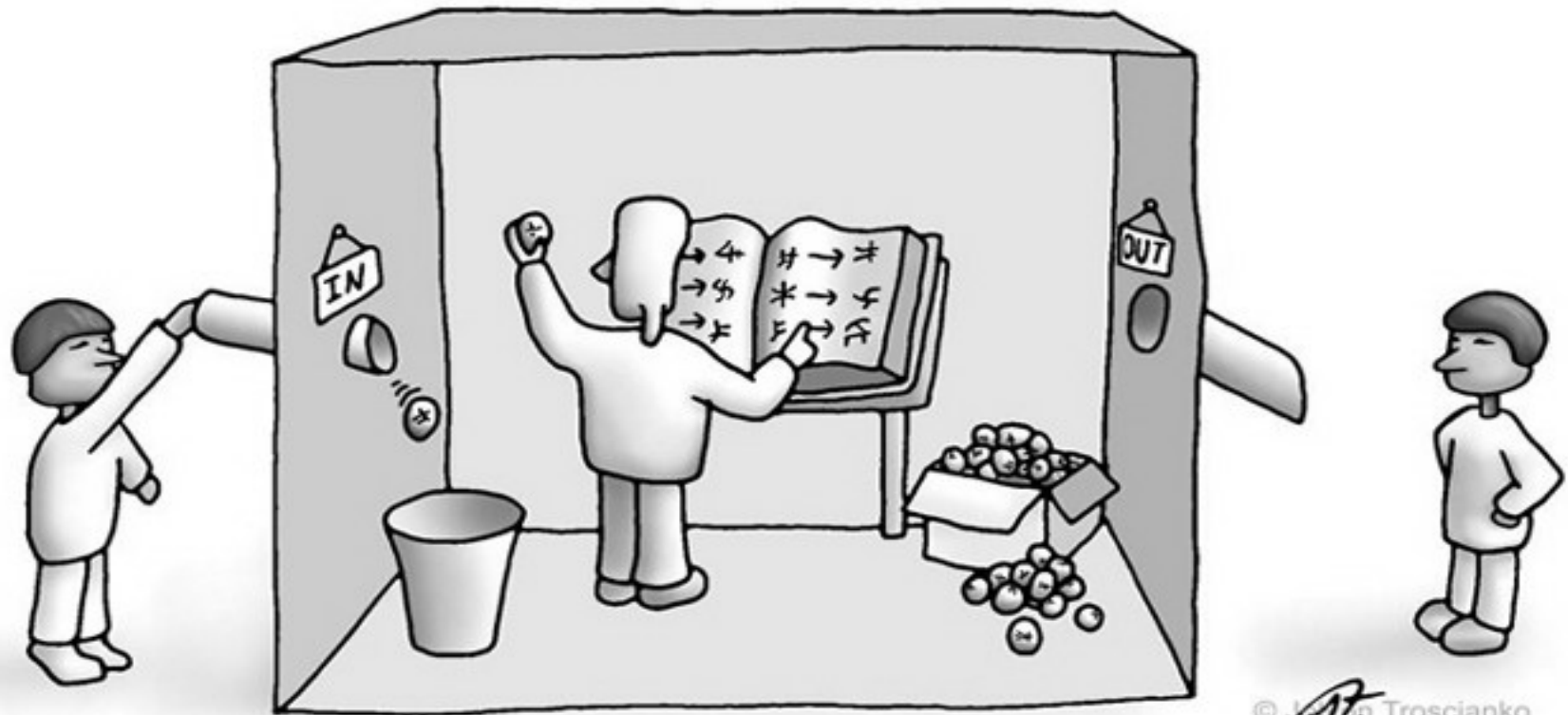
|

Chinese Room

- gedachtenexperiment van John Searle, om kritiek op de Turing test te leveren.
- *stel je voert als een little man in een soort Little Man Computer een gigantisch ingewikkeld programma uit, dat de Turing test doorstaat, maar in het Chinees...*

[J. Searle, "Minds, Brains, and Programs", Behavioral and Brain Sciences, 1980]

Chinese Room



© J. Troscianko
joly.troscianko@lineone.net
From 'Consciousness' S. Blackmore

KI en programmeertalen

- Populaire programmeertalen in het veld van Kunstmatige Intelligentie
 - functionele programmeertalen, mn LISP
 - logische programmeertalen, mn Prolog
- Hele andere manier van programmeren dan de Little Man Computer!

Prolog voorbeeld

Programma bestaat uit predikaten,
bijv. **likes**, met lijst eigenschappen

```
likes(george, kate)
```

```
likes(george, susie)
```

```
likes(george, wine)
```

```
likes(susie, wine)
```

```
likes(kate, gin)
```

```
likes(kate, susie)
```

om dan computer vragen
te laten beantwoorden.

Prolog voorbeeld

Programma bestaat uit predikaten,

bijv. **likes**, met lijst eigenschappen

```
likes(george, kate).
```

```
likes(george, susie).
```

```
likes(george, wine).
```

```
likes(susie, wine).
```

```
likes(kate, gin).
```

```
likes(kate, susie).
```

om dan computer vragen

te laten beantwoorden

```
U: ?- likes(george, kate)
```

```
P: yes
```

```
U: ?- likes(kate, susie)
```

```
P: yes
```

```
U: ?- likes(george, X)
```

```
P: X = kate
```

```
U: ;
```

```
P: X = susie
```

```
U: ;
```

```
P: X = wine
```

```
U: ;
```

```
P: no
```

3x3 paardensprongprobleem in Prolog

1	2	3
4	5	6
7	8	9

Kan een schaakpaard van veld 1 naar veld 3 komen?

3x3 paardensprongprobleem in Prolog

- Specificeer de geldige zetten

```
move(1,6) . move(1,8) . move(2,7) . move(2,9) .  
move(3,4) . move(3,8) . move(4,3) . move(4,9) .  
move(6,7) . move(6,1) . move(7,6) . move(7,2) .  
move(8,3) . move(8,1) . move(9,4) . move(9,2) .
```

1	2	3
4	5	6
7	8	9

- Specificeer paden

```
path(Z,Z) .
```

```
path(X,Y) :- move(X,Z) , path(Z,Y) .
```

- Vraag

```
?- path(1,9) .
```

3x3 paardensprongprobleem in Prolog

- Dit kan mooier (en efficiënter!), door cycles in paden te vermijden.

```
path(Z,Z,L) .
```

```
path(X,Y,L) :- move(X,Z) , not(member(Z,L)) ,  
                path(Z,Y,[Z|L])
```

```
member(X,[X|_]) .
```

```
member(X,[_|_]) :- member(X,_) .
```

- Hier betekent `path(X,Y,L)`

“er is een pad van X naar Y via de posities in lijst L”

