



GÖTEBORGS  
UNIVERSITET

# Vad datorer (och vi?) inte kan svara på

Vetenskapsfestivalen 2022

FREDRIK ENGSTRÖM  
MARTIN KASÅ

# Plan

- Vad är en dator?
- Datorns gränser
- Människans kunskapsgränser
  
- Frågor?



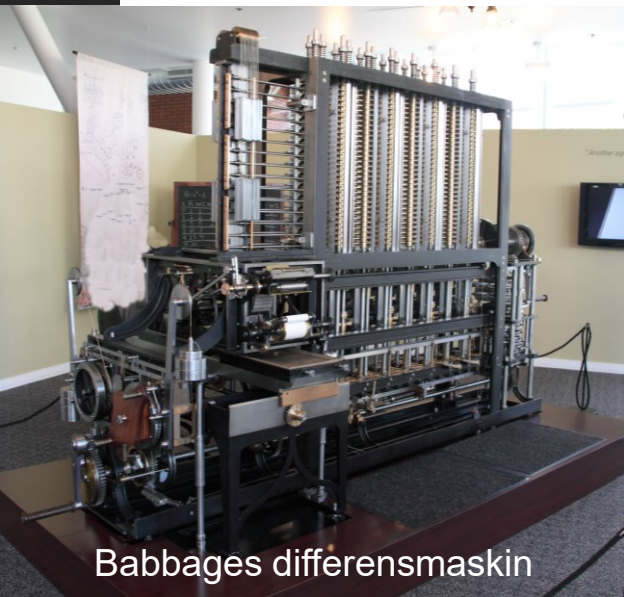
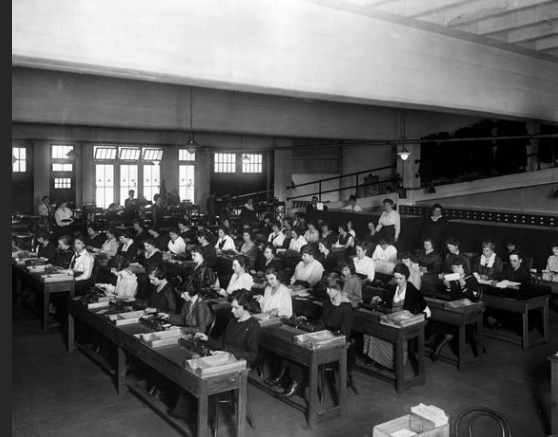


# Vad är en dator?

- En dator är en maskin som utför instruktioner
- Indata processas till utdata
- In- och utdata kan vara textfiler



# Datorer



Babbages differensmaskin



IBM Watson



IBM Quantum System One

# Turingmaskin

Alan Turing 1936

- Obegränsad remsa
- Läs- och skrivhuvud
- Minnescell
- Instruktioner

Allt som går att beräkna kan beräknas med en Turingmaskin.



# Gränser för en dator

Haltproblemet

# Datorprogram som data

- Vilket programspråk är detta program skrivet i?
- Är det ett korrekt skrivet program?
- Hänger sig det här programmet?
- Stannar det här programmet med indata I?

```
neg_result = {}
varval = splitting(elem)
for x in varval_fields:
    if x == "variables" and getattr(varval, x):
        var = getattr(varval, x)
        var = "".join([ACTRVARIABLE, var])
        try:
            temp_result = actrvvariables[var]
        except KeyError:
            raise ACTRError("Object '%s' in the value '%s' is a variable that is not bound; this is illegal in A
    elif x == "values" and getattr(varval, x):
        temp_result = getattr(varval, x)
    elif x == "negvalues" and getattr(varval, x):
        if negative_impossible:
            raise ACTRError("It is not allowed to define negative values or negative variables on the right hand
        else:
            for neg in getattr(varval, x):
                neg_result.add(neg)
    elif x == "negvariables" and getattr(varval, x):
        if negative_impossible:
            raise ACTRError("It is not allowed to define negative values or negative variables on the right hand
        else:
            for neg in getattr(varval, x):
                neg_var = neg
                neg_var = "".join([ACTRVARIABLE, neg_var])
                try:
                    neg_result.add(actrvvariables[neg_var])
                except KeyError:
                    raise ACTRError("Object '%s' in the value '%s' is a variable that is not bound; this is ille
result and temp_result in {VISIONGREATER, VISIONSMALLER} or result in {VISIONGREATER, VISIONSMALLER}:
    result = "".join(sorted([temp_result, result], reverse=True))
if result and temp_result != result:
    raise ACTRError("It looks like in '%s', one slot would have to carry two values at the same time; this is

try:
    result = temp_result
except UnboundLocalError: #temp_result wasn't used
    pass

    _result:
    rned tuple = VarvalClass(variables=None, values=result, negvariables=(), negvalues=tuple(neg_result))
    tuple
    ndLocalError: #temp_result was never used, which means that no values were defined
    tuple = VarvalClass(variables=None, values=None, negvariables=(), negvalues=tuple(neg_result))
    ed_tuple

    lotvals, name1, name2):

    that happen to be tied to the same slots. This function is used in production compilation. dict2

    temp_set, temp_val):

    ts a set of variables and a value (possibly, None), and it returns two dicts.
    s which variables will be substituted by which variable, valued stores which variables will be s
```

# Haltprogrammet

$\text{Halt}(P, I)$  tar två indata:

- $P$  är ett program
- $I$  är vilken indata som helst

$\text{Halt}(P, I)$  ger resultatet "JA" om

- $P$  stannar (efter något antal beräkningssteg) om det startas med indata  $I$

annars resultatet "NEJ".

- Det verkar inte direkt otänkbart att ett sådant program skulle kunna finnas.
- Ett sådant program vore mycket användbart...
- Motsägelseargument
- Låt oss (för resonemangets skull) anta att ett sådant program finns!

# En motsägelse

D(P):

```
if Halt(P,P) = "JA":
```

```
    loop
```

```
else:
```

```
    stop
```

Kör D(D):

- Om  $\text{Halt}(D,D) = \text{"JA"}$  så loopar D(D)
- Om  $\text{Halt}(D,D) = \text{"NEJ"}$  så stannar D(D)

Alltså: Något sådant program  
 $\text{Halt}(P, I)$  finns inte!



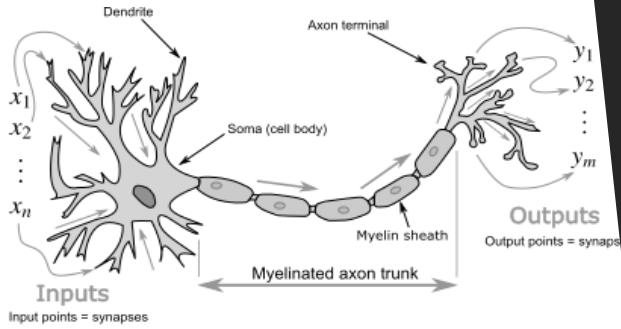
# Slutsats

- Det finns gränser för vad en dator kan åstadkomma.
- Gäller oavsett framsteg inom hårdvara, nya typer av algoritmer eller nya programmeringsspråk.
- Andra problem är också olösbara.

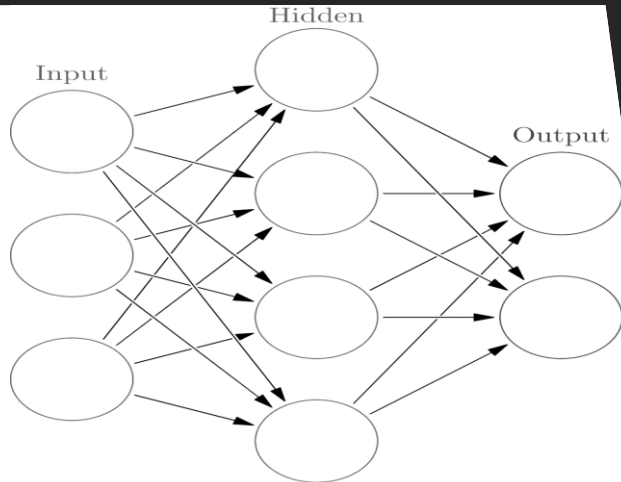
# Människans kunskapsgränser

Är vi bättre än maskiner?

# Människan – en maskin?



- Är människor maskiner?
- Fungerar hjärnan algoritmiskt?
- Kan mänsklighetens problemlösningsförmåga "implementeras" i en Turingmaskin?



- Och vad är kunskap?
- Orakel – finns de?



GÖTEBORGS  
UNIVERSITET

**Fredrik Engström, [fredrik.engstrom@gu.se](mailto:fredrik.engstrom@gu.se)  
Martin Kaså, [martin.kasa@gu.se](mailto:martin.kasa@gu.se)**

**<https://www.gu.se/studera/hitta-utbildning/logik-masterprogram-h2log>**