

Turing Machines

Vraagstelling

Zijn de begrippen algoritme en computer te formaliseren?

Wat is het verband tussen beide?

Welke computer modellen zijn mogelijk?

Is er verschil in rekenkracht?

- geen snelheid maar wat berekend kan worden

Zijn er beperkingen aan wat er überhaupt met een computer berekend kan worden?

Voorbeeld: eindige automaat

Kan gezien worden als een primitieve computer

Maar is beperkt in wat er mee gedaan kan worden

$\{a^i b^i \mid i \geq 0\}$ kan niet

Pushdown automaat

$\{a^i b^i c^i \mid i \geq 0\}$ kan niet

Turing Machine

Turing bedacht zo rond 1935 (voordat er echte computers waren) een computer model gebaseerd op dat van de eindige automaat.

Het blijkt dat dit model universeel is.

- ieder ander computer model dat je kunt verzinnen is niet krachtiger dan een Turing machine
- ieder algoritme dat met een willekeurige computer kan worden uitgevoerd kan ook met een Turing machine worden uitgevoerd

We behandelen e.e.a informeel

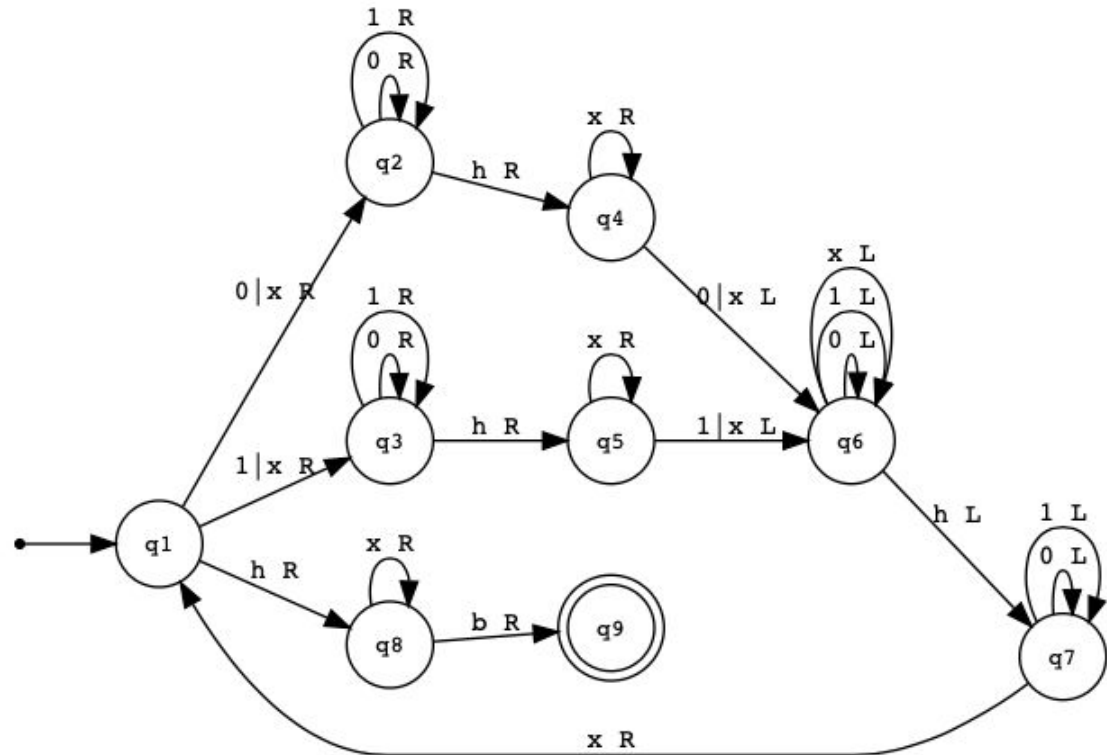
- formele definities zijn online of in boeken te vinden

Turing Machine

- Eindige automaat met als extra een tape met symbolen waarop gelezen en geschreven kan worden
- Leeskop die op bepaalde positie van tape staat
- In iedere state wordt de volgende state bepaald door de combinatie van de state en de waarde onder de leeskop
- Er kan een ander symbool op de plek van de leeskop worden geschreven
- De leeskop gaat een positie naar links of rechts
- Er zijn speciale eindtoestanden (states) waar de berekening stopt
 - Kan een Accept of Reject state zijn
 - Vaak is de inhoud van tape na afloop het antwoord
- Vaak ontbreken de Reject states en is de Reject impliciet als er geen regel toegepast kan worden

Voorbeeld {whwb | $w \in (0|1)^*$ } (b spatie, 110h110b of 11011b11011b)

q1 0 x R q2
 q1 1 x R q3
 q2 0 0 R q2
 q2 1 1 R q2
 q2 h h R q4
 q3 0 0 R q3
 q3 1 1 R q3
 q3 h h R q5
 q4 x x R q4
 q4 0 x L q6
 q5 x x R q5
 q5 1 x L q6
 q6 h h L q7
 q6 0 0 L q6
 q6 1 1 L q6
 q6 x x L q6
 q7 0 0 L q7
 q7 1 1 L q7
 q7 x x R q1
 q1 h h R q8
 q8 x x R q8
 q8 b b R q9
 accept q9
 start q1

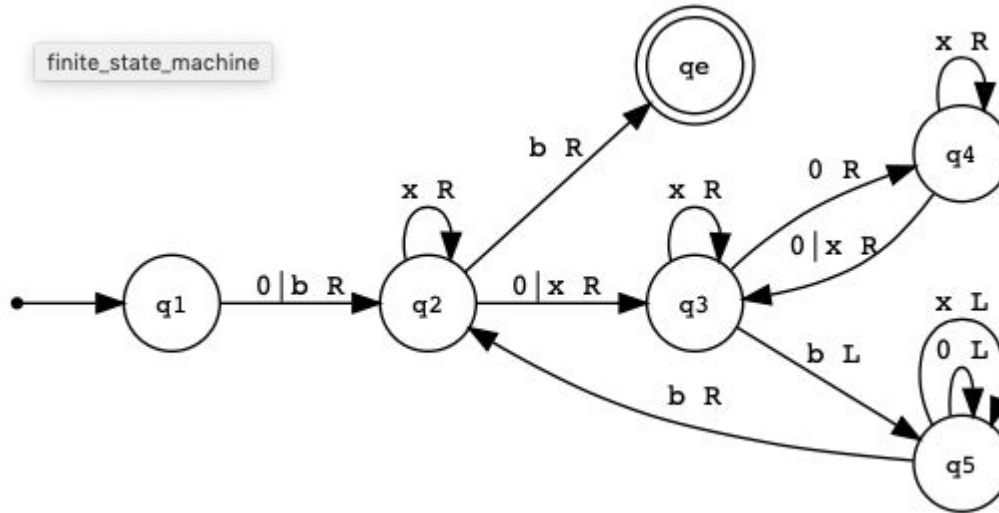


Voorbeeld $\{0^n 2^n \mid n \geq 0\}$ (2^n nullen, spatie op eind, 00b 0000b 00000000b)

```

q1 0 b R q2
q2 b b R qe
q2 x x R q2
q2 0 x R q3
q3 x x R q3
q3 0 0 R q4
q3 b b L q5
q4 x x R q4
q4 0 x R q3
q5 0 0 L q5
q5 x x L q5
q5 b b R q2
start q1
accept qe

```



Turing machine Simuleren in Python

```
def simTuring(tm,tape):
    ts,start,ess,rss = tm
    state = start
    pos = 0
    while state not in ess and state not in rss:
        sym = tape[pos]
        res = [(ws,d,qe) for q,rs,ws,d,qe in ts if state == q and sym == rs]
        if len(res) == 0:
            # print("no rule for: ",state,sym)
            return ('Reject',tape)
        else:
            # print(res[0])
            tape[pos],d,state = res[0]
            pos += 1 if d == 'R' else -1
    if state in ess:
        return ('Accept',tape)
    else:
        return ('Reject',tape)
```


Multitape Turing machine

Turing machine met meerdere tapes met ieder eigen leeskop (gelijktijdig lezen en schrijven)

Iedere multi-tape TM is het een single tape TM te simuleren

Niet deterministische Turing machine

Zelfde als Turing machine maar volgende toestand (en geschreven symbool, etc) niet uniek bepaald door combinatie toestand en symbool.

Voor iedere NTM is er een equivalente DTM (net als bij eindige automaten).

Dit gaat dmv simulatie mbv een multitape machine.

Universele Turing Machine

Is Turing machine waarmee we iedere andere Turing machine kunnen simuleren

Zou een Turing Machine met 2 tapes kunnen zijn:

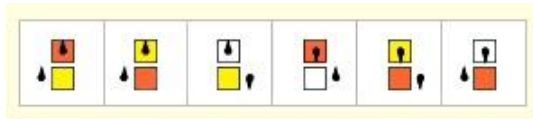
- ene tape bevat programma die een andere Turing machine beschrijft
- andere tape de invoer code voor dit programma

Is dus in feite een gewone computer! Daar staat het programma ook in het geheugen.

Kleinste Universele Turing machine

De [kleinste universele Turing machine](#) heeft twee toestanden (up down) en 3 symbolen (kleuren: white red en yellow).

Plaatje geeft regels weer:



Ieder berekenbaar probleem kan dus naar een input tape bestaande uit w,r,y omgezet worden, waarna de oplossing door deze TN berekend wordt!

u	r	y	L	u
u	y	r	L	u
u	w	y	R	d
d	r	w	R	u
d	y	r	R	d
d	w	r	L	u