

Church-Turing These

Een nieuw paradijs?

Pieter van Engelen

20 juni 2024@Fontys, Sittard



De routeplanner

De these van Church en Turing

Turing machines

One (tu)ring to rule them all

De cirkel

De hoofdrolspelers



Alonzo Church (1903-1995)
Princeton, USA



Alan Turing (1912-1954)
Cambridge/Manchester, UK

Every *effectively calculable* function
is *computable*

Church (1936), Turing (1937)

De termen

Effectively calculable

- Effectief uitrekenbaar
- Zonder intuïtie of interpretatie
- De (reken)regels volgen

Computable

- Formele definitie
- Algoritme



Algoritme

De λ -calculus (Church)

- Vanuit de wiskundige logica
- Algoritme omschreven door wiskundige relaties
- Functioneel programmeren (F#, Haskell, LISP, ...)

Turing machines

- Formalisering van begrip *proces*
- Algoritme omschreven door instructies
- Imperatief programmeren (Python, C, Java, Pascal, ...)

De routeplanner

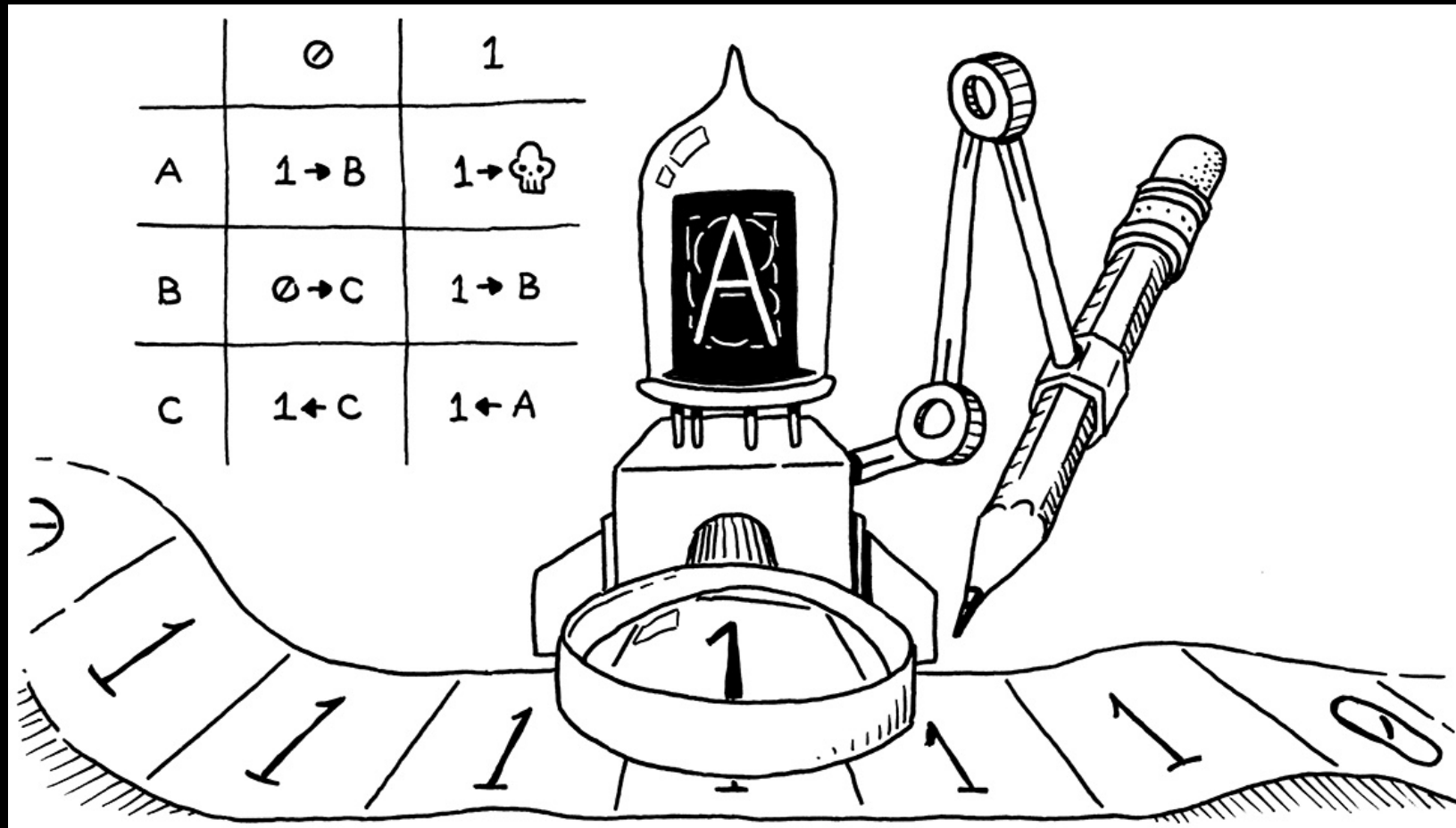
~~De these van Church en Turing~~

Turing machines

One (tu)ring to rule them all

De cirkel

Turing machine

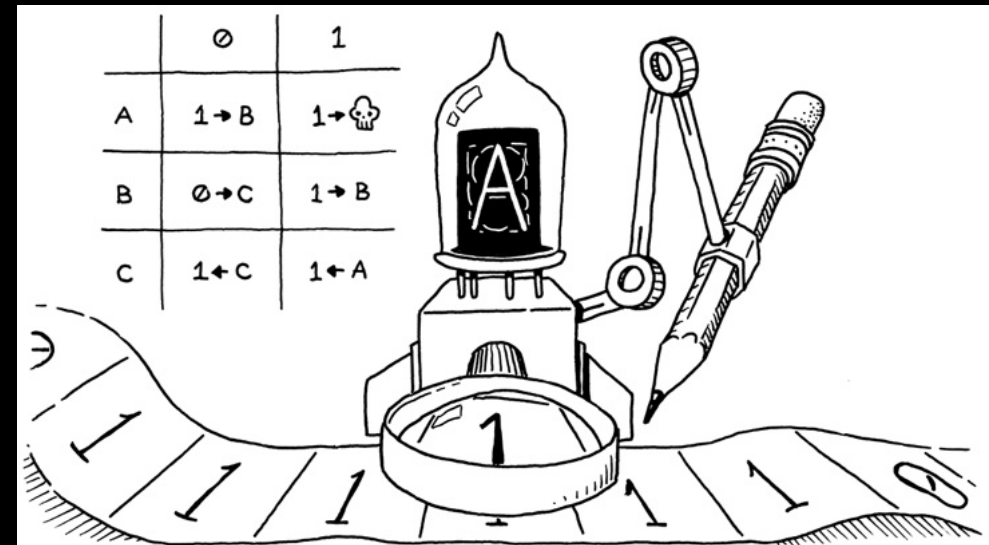


Wat formeler

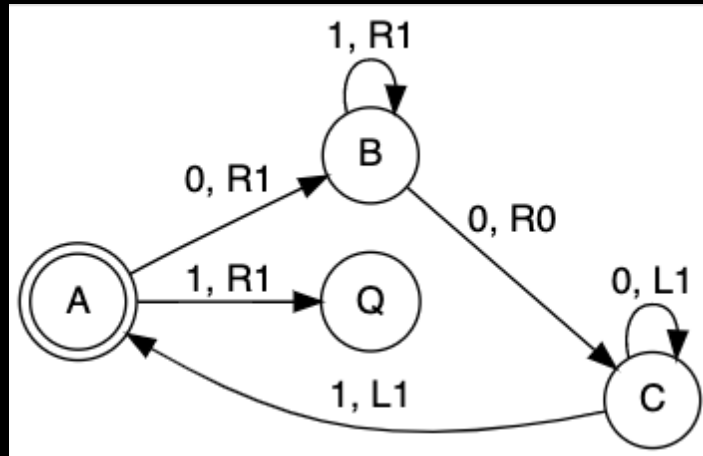
Turing Machine

- Een alfabet Σ van symbolen, bijvoorbeeld $\{a, b\}$ of $\{0, 1, x, c\}$
- Een (oneindige) tape waar symbolen opstaan
- Een eindige verzameling toestanden $q_0, q_1, q_2, \dots, q_n$
- Een eindige verzameling voorschriften
 - Wat staat er op de tape? α
 - Welke kant gaat de leeskop? L of R
 - Wat moet er op de tape komen te staan? β
 - Wat is de nieuwe toestand? q_i
 - A: $0R1B$

Opdracht:
Schrijf de overige voorschriftregels adhv de tabel.



Wat formeler



Tape invoer: 00001

Spoor:

A: 00001

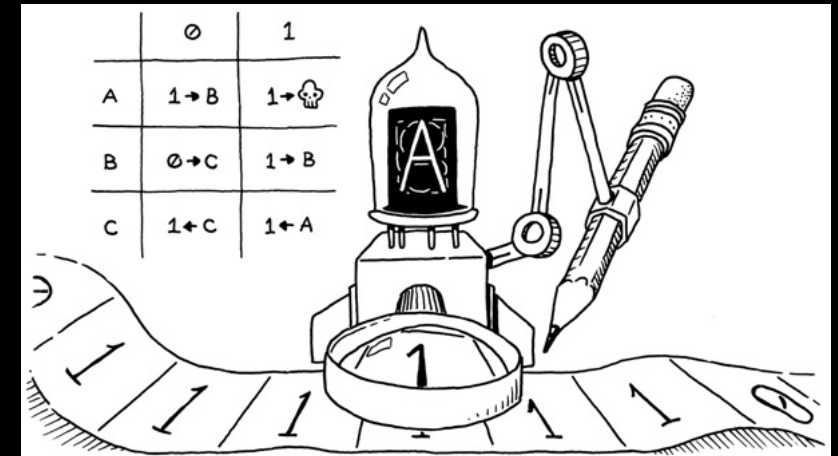
B: 10001

C: 10001

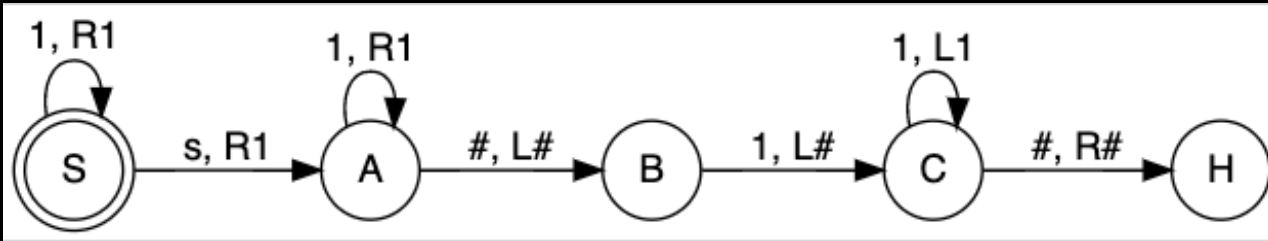
C: 10101

C: 11101

A: 1110

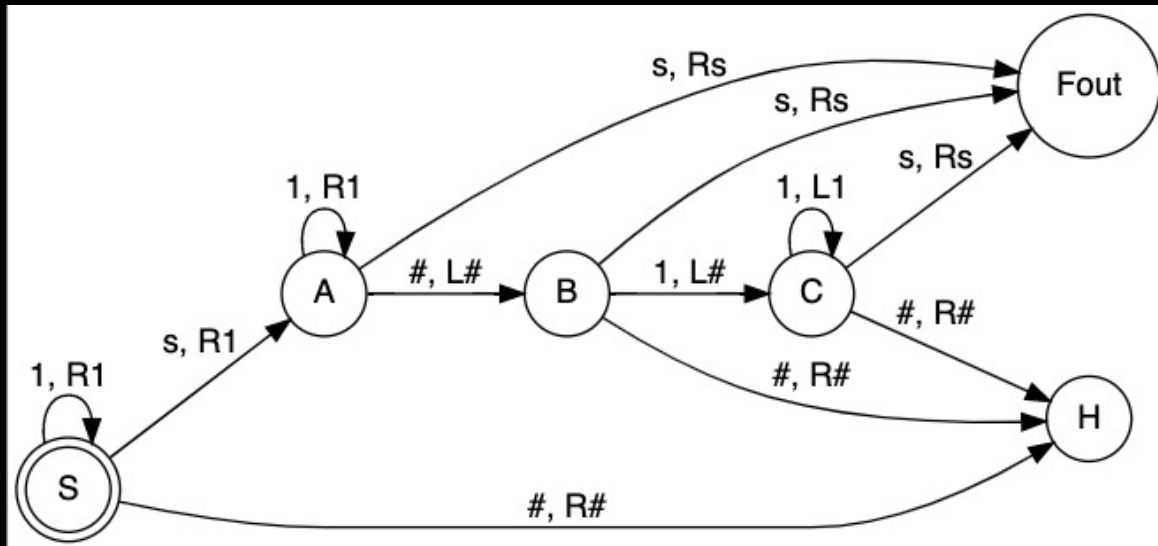


Become a padawan



Opdracht:

Maak een spoor van de tape #1s111#



S: 1R1S

S: sR1A

A: 1R1A

A: #L#B

B: 1L#C

C: 1L1C

C: #R#H

Het spoor:

S: #1s111#

S: #11s111#

S: #11s111#

A: #111111#

A: #111111#

A: #111111#

A: #1111111#

B: #1111111#

C: #111111##

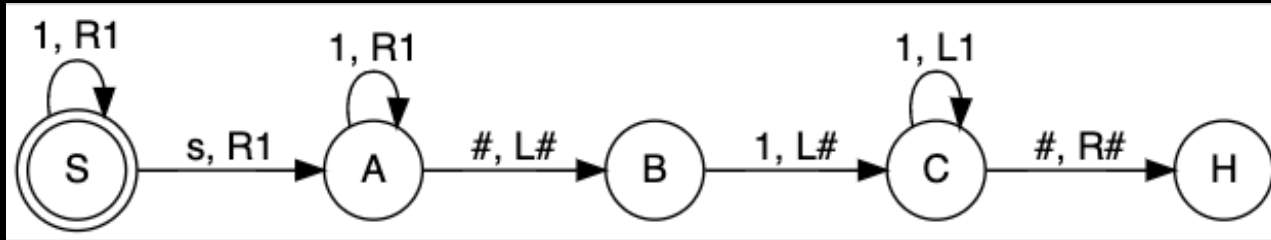
:

C: #111111##

C: #111111##

H: #111111##

Iets van betekenis: unaire optelling



Maak een spoor van de tape #11s111#

www.turingmachine.io

```
input: '11s111 '  
blank: ' '  
start state: start  
table:  
  start:  
  | 1 : {write: 1, R: start}  
  | s : {write: 1, R: A}  
  A:  
  | 1 : {write: 1, R: A}  
  | ' ': {write: ' ', L: B}  
  B:  
  | 1 : {write: ' ', L: C}  
  C:  
  | 1 : {write: 1, L: C}  
  | ' ': {write: ' ', R: halt}  
  halt:
```

Turingmachine in een computer

- Unaire increment
- Unair verschil
- Binaire increment (+1)
- Binaire optelling
- Palindroom herkenning
- String vergelijking

Zelf spelen: www.turingmachine.io

De routeplanner

~~De these van Church en Turing~~

~~Turing machines~~

One (tu)ring to rule them all

De cirkel

Alan Turing (1912-1954)



De belangrijke gebeurtenissen/concepten:

- Turing Machine (1936)
- Breken van de Duitse Enigma-codes mbv machines (1941)
- Bedenker eerste Stored Program Computer (1946)
- Grondlegger van AI
- Computing Machinery and Intelligence (1950)
- Grondlegger Morphogenesis (1952)
- Bijna deelnemer Olympische Spelen 1948
- Veroordeeld om zijn homoseksualiteit (1952)
- Gestorven onder verdachte omstandigheden

One (tu)ring to rule them all

Probleem: per functie een aparte Turing machine maken

Oplossing: de Universele Turing Machine (1936)

Hoe?

- Codeer alle definiërende eigenschappen in symbolen → een TM kan op tape gezet worden!
- Maak een voorschrift hoe met de voorschriften en deze gecodeerde TM omgegaan moet worden (serieus ingewikkeld, maar te doen)

Gevolgen:

- We kunnen redeneren over alle Turing Machines *en* hun gedrag!
- We kennen vanaf dit moment het begrip *Berekenbaarheid*.

One (tu)ring to rule them all

Belangrijke resultaten:

- Er bestaat geen voorschrift om binnen eindige tijd te beslissen of een Turing Machine ooit 'stopt'

Het zgn *Halting Problem*

- Er bestaat geen voorschrift om binnen eindige tijd te beslissen of een Turing Machine ooit een gegeven symbool op de tape zet.

Deze is belangrijk! 😊

- Turing machines zijn de 'eerste' klasse van machines, die zichzelf kunnen 'nadoen'.
- De eerste orde logica en de Peano-rekenkunde zijn te modelleren in TM's

De routeplanner

~~De these van Church en Turing~~

~~Turing machines~~

~~One (tu)ring to rule them all~~

De cirkel

DIE GRUNDLEHREN DER
MATHEMATISCHEN
WISSENSCHAFTEN

IN EINZELDARSTELLUNGEN MIT BESONDERER
BERÜCKSICHTIGUNG DER ANWENDUNGSGEBIETE

HERAUSGEGEBEN VON

G. D. BIRKHOFF · W. BLASCHKE · R. COURANT
R. GRAMMEL · M. MORSE · F. K. SCHMIDT
B. L. VAN DER WAERDEN

BAND XXVII

GRUNDZÜGE DER THEORETISCHEN LOGIK

VON

D. HILBERT UND W. ACKERMANN

ZWEITE AUFLAGE

GRUNDZÜGE
DER THEORETISCHEN
LOGIK

VON

D. HILBERT

GEHEIMER REGIERUNGSRAT
PROFESSOR AN DER UNIVERSITÄT GÖTTINGEN

UND

W. ACKERMANN

BÜRGSTEINFURT

ZWEITE, VERBESSERTE AUFLAGE

PUBLISHED AND DISTRIBUTED IN THE PUBLIC INTEREST
BY AUTHORITY OF THE U.S. ALIEN PROPERTY CUSTODIAN
UNDER LICENSE NO. A-695

De cirkel

De cirkel

Entscheidungsproblem in de eerste orde predicaatlogica (Hilbert & Ackermann, 1928)

*Bepaal volgens een stappenplan/proces
of een uitspraak waar of niet waar is.*

Alan Turing en Alonzo Church allebei in 1936, onafhankelijk van elkaar:

“Kan niet. Kijk maar. Sorry, David.”

On Computable Numbers, with an Application to the Entscheidungsproblem, A. Turing,
Proceedings of the London Mathematical Society 2 (42): 230--265 (1936)

An Unsolvability Problem of Elementary Number Theory, A. Church,
American Journal of Mathematics Vol. 58, No. 2 (Apr., 1936), pp. 345-363

Algoritme

De λ -calculus (Church)

- Vanuit de wiskundige logica
- Algoritme omschreven door wiskundige relaties
- Functioneel programmeren (F#, Haskell, LISP, ...)

Turing machines

- Formalisering van begrip *proces*
- Algoritme omschreven door instructies
- Imperatief programmeren (Python, C, Java, Pascal, ...)

Bij elk λ -programma kan ik een turingmachine maken.

Turing (1937)

De routeplanner

~~De these van Church en Turing~~

~~Turing machines~~

~~One (tu)ring to rule them all~~

~~De cirkel~~

Samenvattend

Church en Turing werkten onafhankelijk
tegelijk aan hetzelfde

Entscheidungsproblem

Halting problem

Grondslagen van de informatica

De geboorte van de 'papieren' computer

Zijn er nog vragen?

Dankjewel voor je aandacht

Kijk vanaf vanavond op pietervanengelen.nl voor de recap van dit college.