



# Basismodellen in de Informatica

## Voorlichting TW

Jaco van de Pol  
Arend Rensink

Universiteit Twente, EWI, Formele Methoden en Tools

- ▶ Motivatie
  - ▶ Rekenen en Taal
  - ▶ basismodellen – basisvragen
- ▶ Basismodellen: sneak preview
  - ▶ Automaten
  - ▶ Grammatica's
  - ▶ Turing machines
- ▶ Vervolg
  - ▶ Toepassingen
  - ▶ Vakken
- ▶ Organisatie



- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?



- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???



- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???
  - ▶ Rekenen !!!
- ▶ Computer Science

- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???
  - ▶ Rekenen !!!
- ▶ Computer Science
- ▶ Wat is **rekenen**?
  - ▶ manipuleren met getallen (numerieke wiskunde)
  - ▶ manipuleren met symbolen (Matlab/Mathematica, wiskundige-stellingbewijzers)

- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???
  - ▶ Rekenen !!!
- ▶ Computer Science
- ▶ Wat is **rekenen**?
  - ▶ manipuleren met getallen (numerieke wiskunde)
  - ▶ manipuleren met symbolen (Matlab/Mathematica, wiskundige-stellingbewijzers)
- ▶ Wat heeft dat met **taal** te maken?
  - ▶ **woord**: een rijtje symbolen
  - ▶ **taal**: een verzameling woorden

- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???
  - ▶ Rekenen !!!
- ▶ Computer Science
- ▶ Wat is **rekenen**?
  - ▶ manipuleren met getallen (numerieke wiskunde)
  - ▶ manipuleren met symbolen (Matlab/Mathematica, wiskundige-stellingbewijzers)
- ▶ Wat heeft dat met **taal** te maken?
  - ▶ **woord**: een rijtje symbolen
  - ▶ **taal**: een verzameling woorden
  - ▶ **Vb**: Natuurlijke talen (**Fries**);

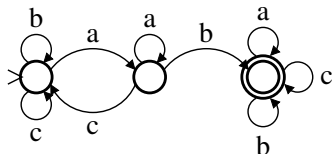
- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???
  - ▶ Rekenen !!!
- ▶ Computer Science
- ▶ Wat is **rekenen**?
  - ▶ manipuleren met getallen (numerieke wiskunde)
  - ▶ manipuleren met symbolen (Matlab/Mathematica, wiskundige-stellingbewijzers)
- ▶ Wat heeft dat met **taal** te maken?
  - ▶ **woord**: een rijtje symbolen
  - ▶ **taal**: een verzameling woorden
  - ▶ **Vb**: Natuurlijke talen (**Fries**); Programmeertalen (**Java**)

- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???
  - ▶ Rekenen !!!
- ▶ Computer Science
- ▶ Wat is **rekenen**?
  - ▶ manipuleren met getallen (numerieke wiskunde)
  - ▶ manipuleren met symbolen (Matlab/Mathematica, wiskundige-stellingbewijzers)
- ▶ Wat heeft dat met **taal** te maken?
  - ▶ **woord**: een rijtje symbolen
  - ▶ **taal**: een verzameling woorden
  - ▶ **Vb**: Natuurlijke talen (**Fries**); Programmeertalen (**Java**)Expressies:  $(a + b) \times c$ ;

- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???
  - ▶ Rekenen !!!
- ▶ Computer Science
- ▶ Wat is **rekenen**?
  - ▶ manipuleren met getallen (numerieke wiskunde)
  - ▶ manipuleren met symbolen (Matlab/Mathematica, wiskundige-stellingbewijzers)
- ▶ Wat heeft dat met **taal** te maken?
  - ▶ **woord**: een rijtje symbolen
  - ▶ **taal**: een verzameling woorden
  - ▶ **Vb**: Natuurlijke talen (**Fries**); Programmeertalen (**Java**)  
Expressies:  $(a + b) \times c$ ; Stellingen:  $\forall c, n. \exists a, b. a^n + b^n = c^n$

- ▶ Wat is het belangrijkste **werkwoord** in de Informatica?
  - ▶ Programmeren ???
  - ▶ Rekenen !!!
- ▶ Computer Science
- ▶ Wat is **rekenen**?
  - ▶ manipuleren met getallen (numerieke wiskunde)
  - ▶ manipuleren met symbolen (Matlab/Mathematica, wiskundige-stellingbewijzers)
- ▶ Wat heeft dat met **taal** te maken?
  - ▶ **woord**: een rijtje symbolen
  - ▶ **taal**: een verzameling woorden
  - ▶ **Vb**: Natuurlijke talen (**Fries**); Programmeertalen (**Java**)  
Expressies:  $(a + b) \times c$ ; Stellingen:  $\forall c, n. \exists a, b. a^n + b^n = c^n$
- ▶ Basismodellen sluit het meest aan op **Discrete Wiskunde**

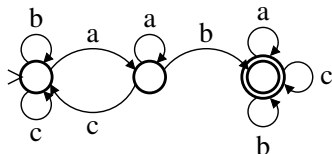
- ▶ Graaf met **toestanden** en **pijlen** met **symbolen**
- ▶ Een woord wordt **herkend** als er een pad is van begin naar eind
- ▶ Hier: Woorden over  $\{a, b, c\}^*$  die het subwoord **ab** bevatten



Alternatief model: Reguliere expressies

$(a+b+c)^*.a.b.(a+b+c)^*$

- ▶ Graaf met **toestanden** en **pijlen** met **symbolen**
- ▶ Een woord wordt **herkend** als er een pad is van begin naar eind
- ▶ Hier: Woorden over  $\{a, b, c\}^*$  die het subwoord **ab** bevatten



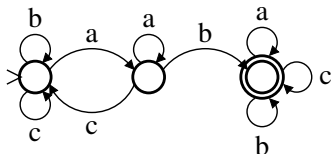
Alternatief model: Reguliere expressies

$(a+b+c)^*.a.b.(a+b+c)^*$

Het machine model “eindige automaten” is **fundamenteel beperkt**:

- ▶ Veel talen kunnen door geen enkele automaat herkend worden:

- ▶ Graaf met **toestanden** en **pijlen** met **symbolen**
- ▶ Een woord wordt **herkend** als er een pad is van begin naar eind
- ▶ Hier: Woorden over  $\{a, b, c\}^*$  die het subwoord **ab** bevatten



Alternatief model: Reguliere expressies

$(a+b+c)^*.a.b.(a+b+c)^*$

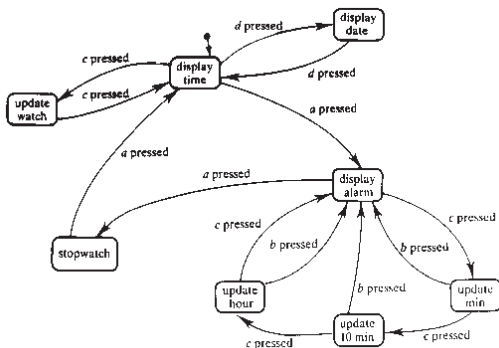
Het machine model “eindige automaten” is **fundamenteel beperkt**:

- ▶ Veel talen kunnen door geen enkele automaat herkend worden:
  - ▶ Palindromen. Vb: *aba*, *baaabbaaab*,  
**Baas, neem een racecar, neem een Saab**



## Modelleren van Discrete Systemen (**digitaal horloge**):

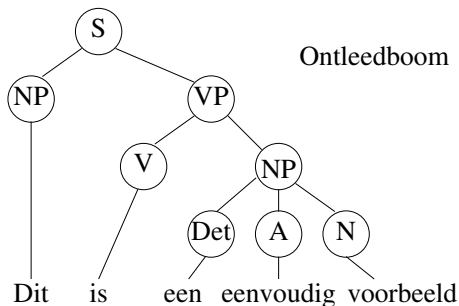
- ▶ **symbol**: event/interactie met omgeving (druk knop / display)
- ▶ **woord**: een specifieke executie van het systeem
- ▶ **taal**: totale systeemgedrag (verzameling van alle executies)



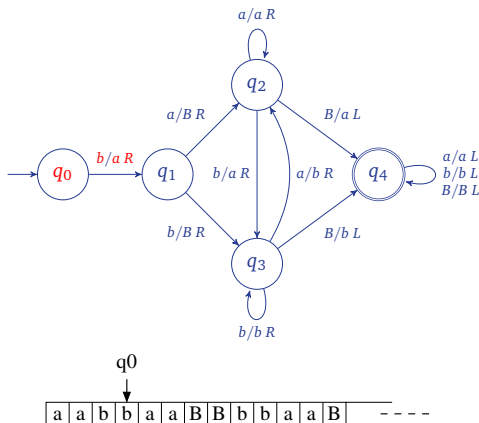


Regels		Woordenboek	
S	→ NP VP	Det	→ een
NP	→ Det A N	N	→ voorbeeld
VP	→ V	A	→ eenvoudig
VP	→ V NP	V	→ is
		NP	→ dit

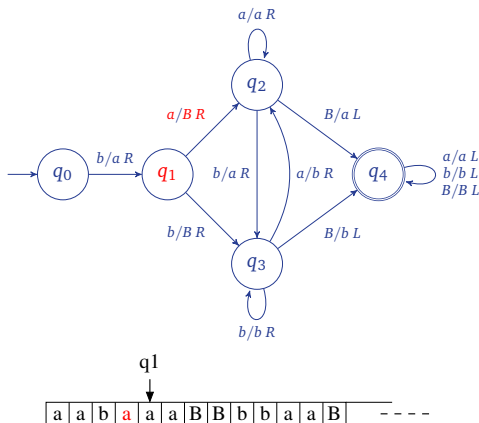
Regels	Woordenboek
$S \rightarrow NP VP$	Det $\rightarrow$ een
$NP \rightarrow Det A N$	N $\rightarrow$ voorbeeld
$VP \rightarrow V$	A $\rightarrow$ eenvoudig
$VP \rightarrow V NP$	V $\rightarrow$ is
	NP $\rightarrow$ dit



- ▶ Eindige Automaat (besturing)
- ▶ Onbegrensde Tape (geheugen)



- ▶ Eindige Automaat (besturing)
- ▶ Onbegrensde Tape (geheugen)





Alan Turing (1912–1954)  
wiskundige, “informaticus”  
kraker van de Enigma code

Turing Machine

Turing Test



Noam Chomsky (1928 –)  
taalkundige,  
politiek anarchist

Universele Grammatica

Chomsky Hierarchy

- ▶ Hoe **expressief** zijn de verschillende machine-modellen?  
(wat kunnen ze berekenen = welke talen kunnen ze herkennen)
  - ▶ eindige automaten < grammatica's < Turing Machines
  - ▶ **eindige automaten** zijn even expressief als **reguliere expressies**

- ▶ Hoe **expressief** zijn de verschillende machine-modellen?  
(wat kunnen ze berekenen = welke talen kunnen ze herkennen)
  - ▶ eindige automaten < grammatica's < Turing Machines
  - ▶ **eindige automaten** zijn even expressief als **reguliere expressies**
- ▶ Wat zijn de **fundamentele grenzen** aan berekenbaarheid?
  - ▶ **Hypothese**: Een Turing M is even krachtig als elke computer
  - ▶ Om te bewijzen dat een probleem **onberekenbaar** is, hoef je dus alleen maar aan te tonen dat er geen TM voor bestaat!

- ▶ Hoe **expressief** zijn de verschillende machine-modellen?  
(wat kunnen ze berekenen = welke talen kunnen ze herkennen)
  - ▶ eindige automaten < grammatica's < Turing Machines
  - ▶ **eindige automaten** zijn even expressief als **reguliere expressies**
- ▶ Wat zijn de **fundamentele grenzen** aan berekenbaarheid?
  - ▶ **Hypothese**: Een Turing M is even krachtig als elke computer
  - ▶ Om te bewijzen dat een probleem **onberekenbaar** is, hoef je dus alleen maar aan te tonen dat er geen TM voor bestaat!
- ▶ Hoe gemakkelijk zijn diverse modellen te **analyseren**?
  - ▶ Zwakke modellen zijn veel beter te analyseren dan krachtige
  - ▶ Voor Turing Machines zijn de meeste vragen **onbeslisbaar**
  - ▶ De vraag of een stelling waar is, is ook **onbeslisbaar**
  - ▶ **Een wiskundige is dus niet te automatiseren**

- ▶ **Programmeertalen:** Theorie en Praktijk
  - ▶ Vertalerbouw (B2-211035)

- ▶ **Programmeertalen:** Theorie en Praktijk
  - ▶ Vertalerbouw (B2-211035)
- ▶ **Redeneren** over Programma's en Systemen (**correctheid**)
  - ▶ Formal Methods and Software Engineering (B2-213520)
  - ▶ Modeling and Analysis of Computer Systems (M 213531)
  - ▶ Performance Evaluation (M 262030)

- ▶ **Programmeertalen:** Theorie en Praktijk
  - ▶ Vertalerbouw (B2-211035)
- ▶ **Redeneren** over Programma's en Systemen (**correctheid**)
  - ▶ Formal Methods and Software Engineering (B2-213520)
  - ▶ Modeling and Analysis of Computer Systems (M 213531)
  - ▶ Performance Evaluation (M 262030)
- ▶ **Theorie** van Berekenen
  - ▶ Algorithms, datastructures and complexity (B2-214020)
  - ▶ **Computability and Computational Complexity** (M 211170)
  - ▶ Gerelateerd: Mathematical Programming, Graph Theory (B3-TW)

- ▶ **Programmeertalen:** Theorie en Praktijk
  - ▶ Vertalerbouw (B2-211035)
- ▶ **Redeneren** over Programma's en Systemen (**correctheid**)
  - ▶ Formal Methods and Software Engineering (B2-213520)
  - ▶ Modeling and Analysis of Computer Systems (M 213531)
  - ▶ Performance Evaluation (M 262030)
- ▶ **Theorie** van Berekenen
  - ▶ Algorithms, datastructures and complexity (B2-214020)
  - ▶ **Computability and Computational Complexity** (M 211170)
  - ▶ Gerelateerd: Mathematical Programming, Graph Theory (B3-TW)
- ▶ **Natuurlijke taal** verwerking
  - ▶ Speech and language processing (B3-216631)
  - ▶ **Sinds 1950** probeert men al **Russisch-Engels te vertalen!**

- ▶ 8x hoorcollege
  - ▶ Theorie wordt uitgelegd.
  - ▶ Boek: T.A. Sudkamp, **Languages and Machines**  
– An Introduction to the Theory of Computer Science
  - ▶ Docenten: Van de Pol, Rensink
- ▶ 8x werkcollege
  - ▶ Oefenen met opgaven op papier
  - ▶ (Bijna) noodzakelijk om het tentamen te halen
- ▶ 8x praktikum
  - ▶ Graphische tools om constructies op automaten etc. te oefenen
  - ▶ Programmeeropgaven (met vrijstelling voor TW studenten)
  - ▶ **NEW:** TW studenten doen praktikum in de Citadel

## Basismodellen in de Informatica

- ▶ is een boeiend en intrigerend vak:
  - ▶ leuke puzzels
  - ▶ fundamentele vragen
  
- ▶ is een noodzakelijk vak
  - ▶ voor de “wiskunde van rekenen en taal”
  - ▶ **nuttige voorkennis** voor heel veel andere Informatica vakken