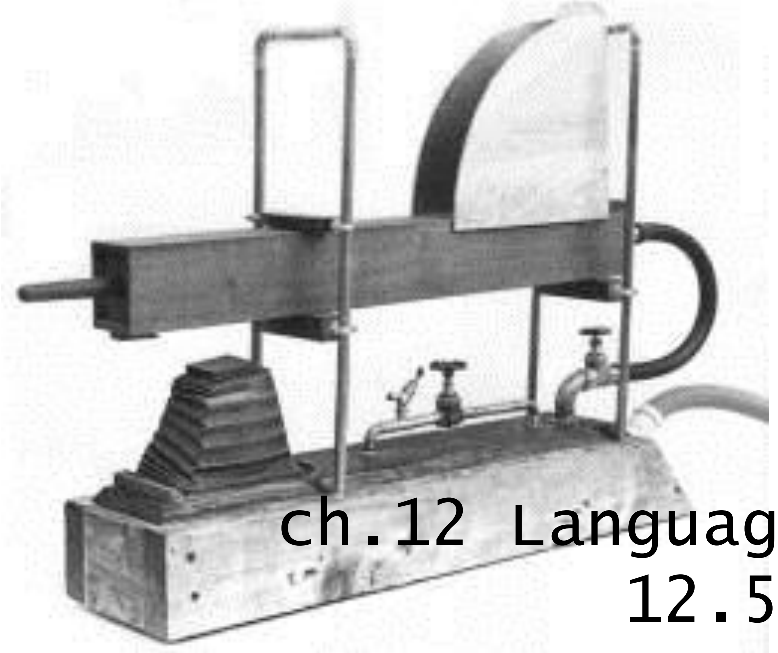
Eindige Automaten

12



Dertiende college

zie dictaatje 4.2

ch.12 Languages, Automata, Grammars
12.5 finite state automata

toestand-actie-diagrammen

Eindige automaten zijn voorbeelden van zgn. toestand-actie-diagrammen. Vanuit een toestand raak je via een actie in een andere toestand. “Eindig” omdat er slechts eindig veel mogelijke toestanden en acties zijn.
De diagrammen zijn handig om systemen te analyseren en/of te modelleren. We formaliseren het “gedrag” van een systeem als de bijbehorende taal van actie-reeksen.

Eerst voorbeelden.

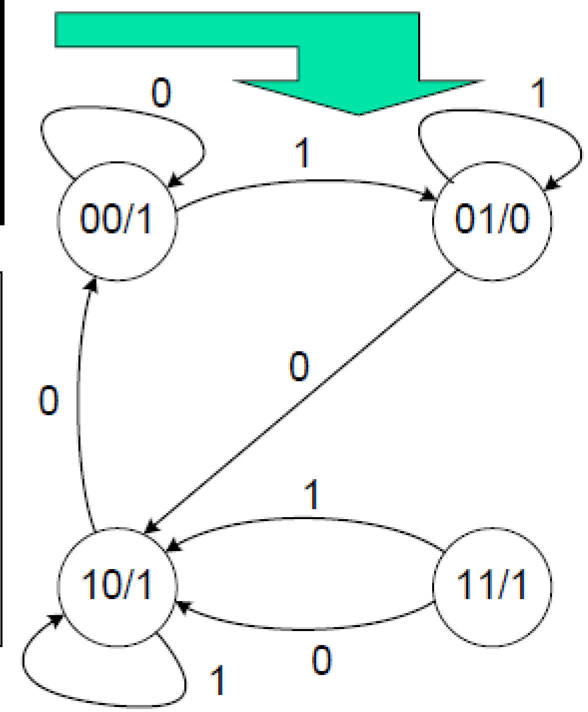
voorbeelden

- de state-transition diagrams van **DiTe** zijn technisch anders maar hebben een gelijke onderliggende filosofie.
- bij **Algoritmiek** kun je problemen/puzzels modelleren door de mogelijke toestanden te inventariseren. Dit kan helpen bij het vinden van een oplossing.
- abstracte beschrijving van het gedrag van –bijvoorbeeld- (de controller van) een automatische deur, lift, vaatwasser, ….
- **syntax-diagrammen** voor stukjes programmeertaal hebben ook toestanden en overgangen.
- eindige automaten voor het herkennen/ van een formele taal: er zijn accepterende toestanden.

Example: Moore Machine (cont.)

Inputs	Present State		Next State		Outputs
x(t)	$q_1(t)$	$q_2(t)$	$q_1(t+1)$	$q_2(t+1)$	z(t)
0	0	0	0	0	1
0	[illegible]	[illegible]	[illegible]	0	0
[illegible]	[illegible]	[illegible]	[illegible]	0	1
[illegible]	[illegible]	[illegible]	[illegible]	0	1
[illegible]	[illegible]	[illegible]	[illegible]	1	1
[illegible]	[illegible]	[illegible]	[illegible]	1	0
[illegible]	[illegible]	[illegible]	[illegible]	0	1
[illegible]	[illegible]	[illegible]	[illegible]	0	1

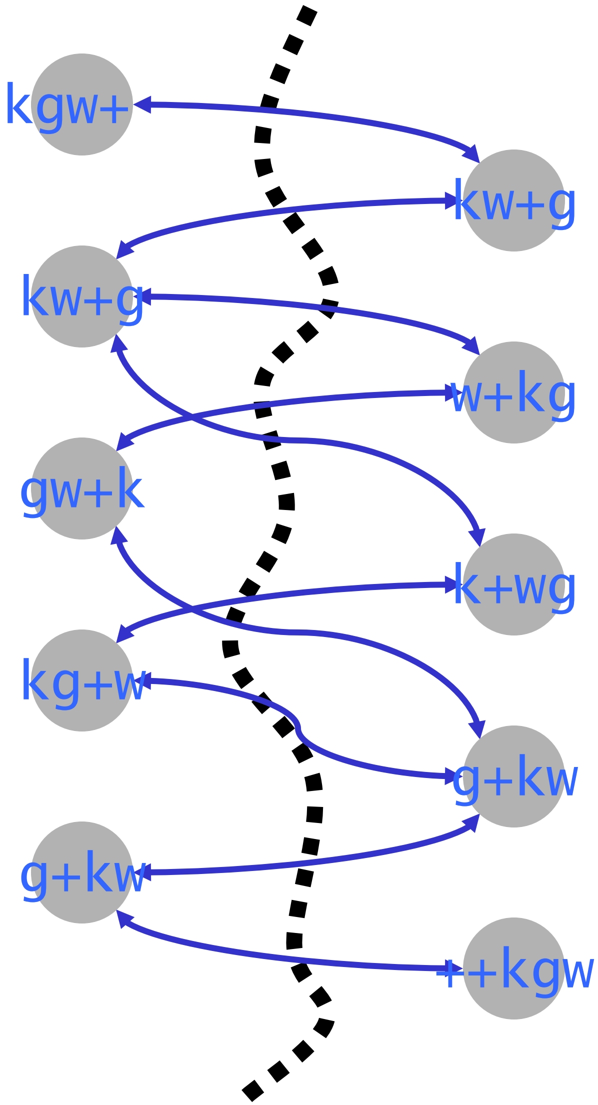
- Possible states = { 00, 01, 10, 11 } → 4 nodes in the diagram
- Possible Transitions = #rows in table → 8 edges in the diagram

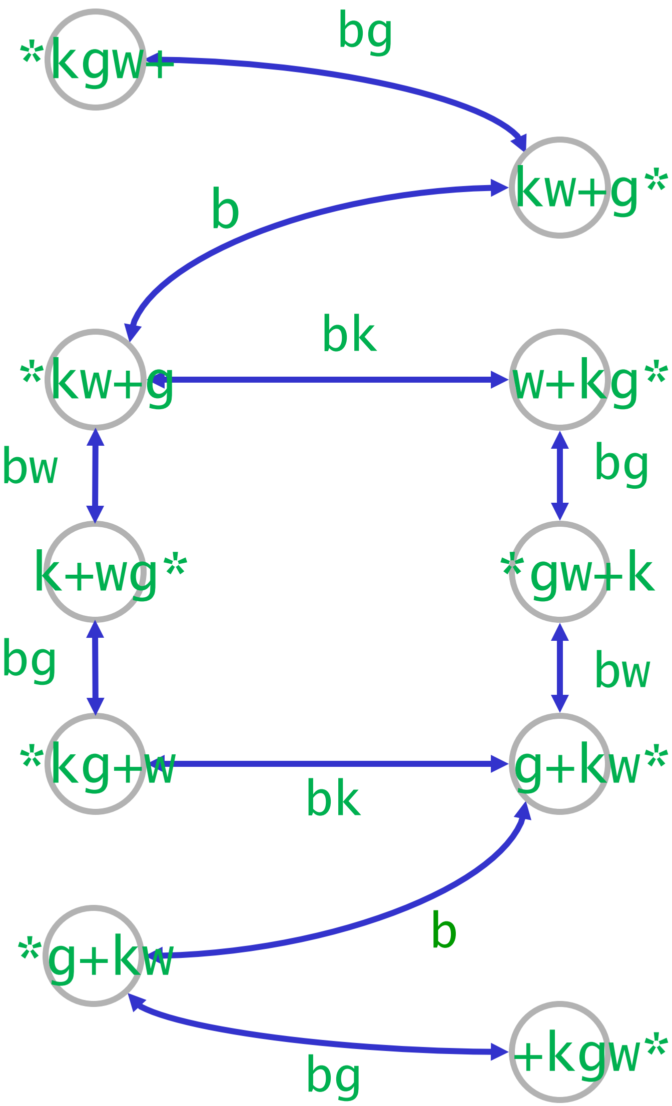



```
state / transition
toestand / actie
maar:
• 'interne bits'
• uitvoer
• (non)determinisme
• begin- en eindtoestanden
• taal
```

Reads as:
When at state ***Q1*** with output ***Z1*** and apply input ***X***, we proceed to state ***Q2*** with output ***Z2.***

kool, geit, wolf

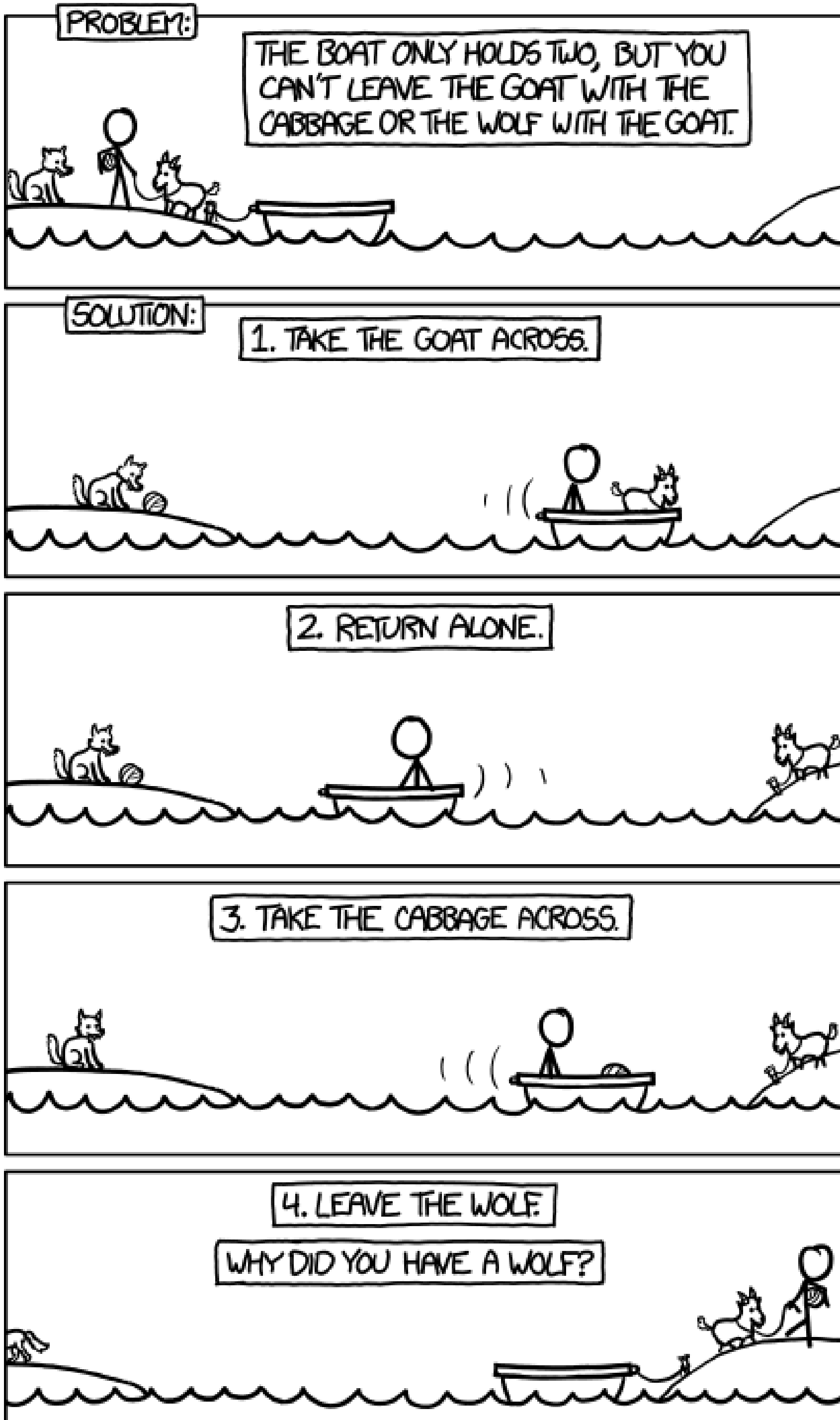


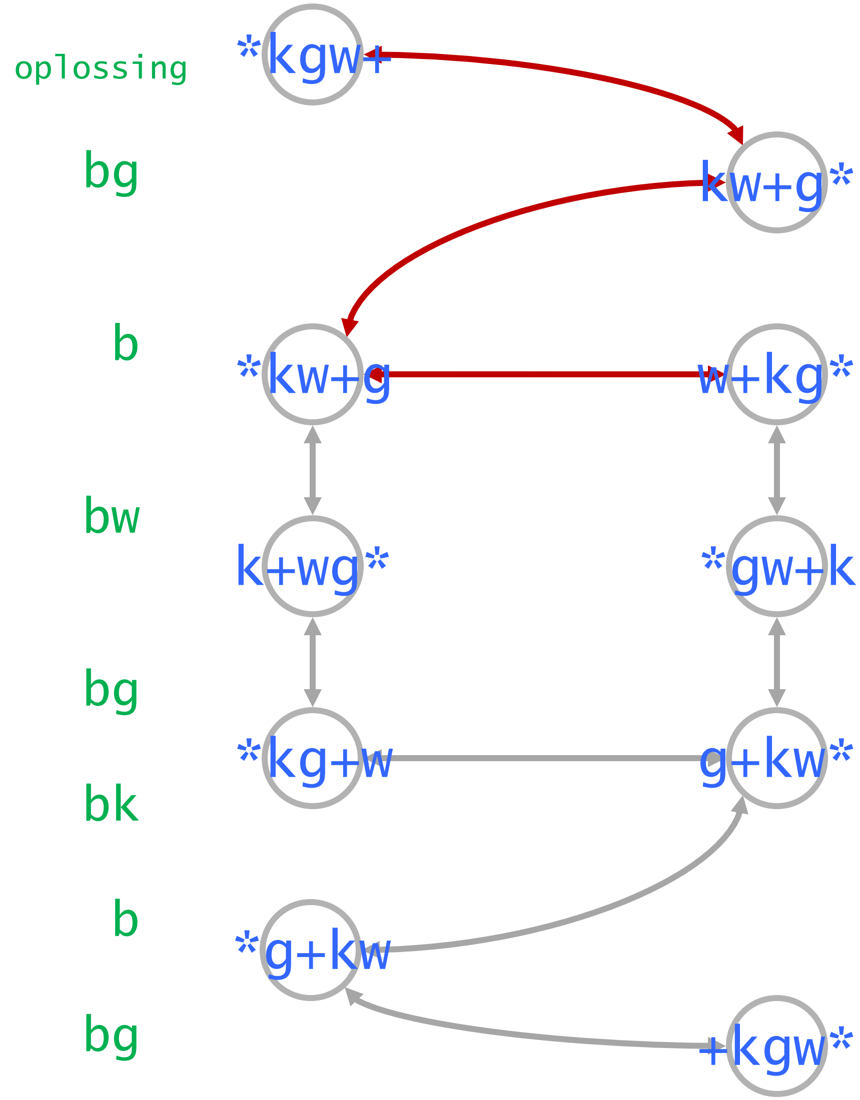


*: boer met boot

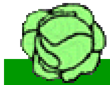
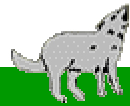
algoritmiek

Logic Boat





http://xkcd.com/1134/

Propositiones ad Acuendos Juvenes

Alcuinus van York (York ~735 - Tours 804)

XVIII. PROPOSITIO DE HOMINE ET CAPRA ET LVPO.

Homo quidam debebat ultra fluvium transferre lupum, capram, et fasciculum cauli. Et non potuit aliam nauem inuenire, nisi quae duos tantum ex ipsis ferre ualebat. Praeceptum itaque ei fuerat, ut omnia haec ultra illaesa omnino transferret. Dicat, qui potest, quomodo eis illaesis transire potuit?

Solutio
Simili namque tenore ducerem prius capram et dimitterem foris lupum et caulum. Tum deinde uenirem, lupumque transferrem: lupoque foris misso capram naui receptam ultra reducerem; capramque foris missam caulum transueherem ultra; atque iterum remigassem, capramque assumptam ultra duxissem. Sicque faciendo facta erit remigatio salubris, absque uoragine lacerationis.

http://en.wikipedia.org/wiki/Propositiones_ad_acuendos_juvenes

automatische deur

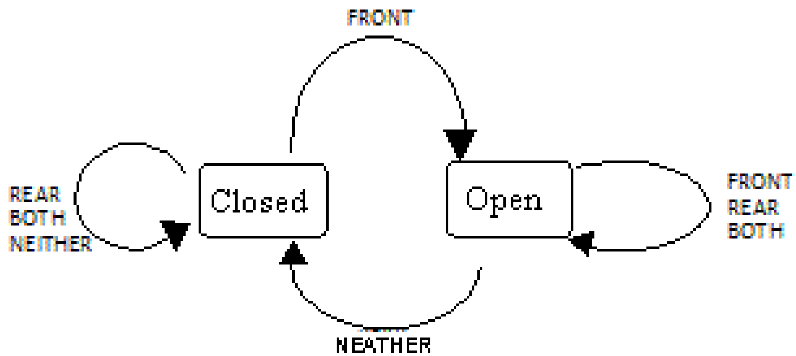


Figure 2: State Transition Diagram

Automatic door controller

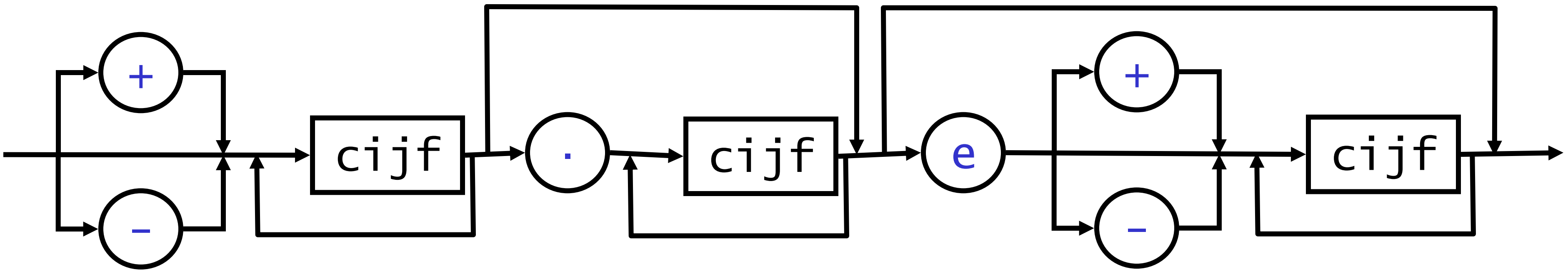
Given state of door	Shopper location			
	Neither	Front	Rear	Both
	What will happen to door (Next State)			
Closed	Closed	Open	Closed	Closed
Open	Closed	Open	Open	Open

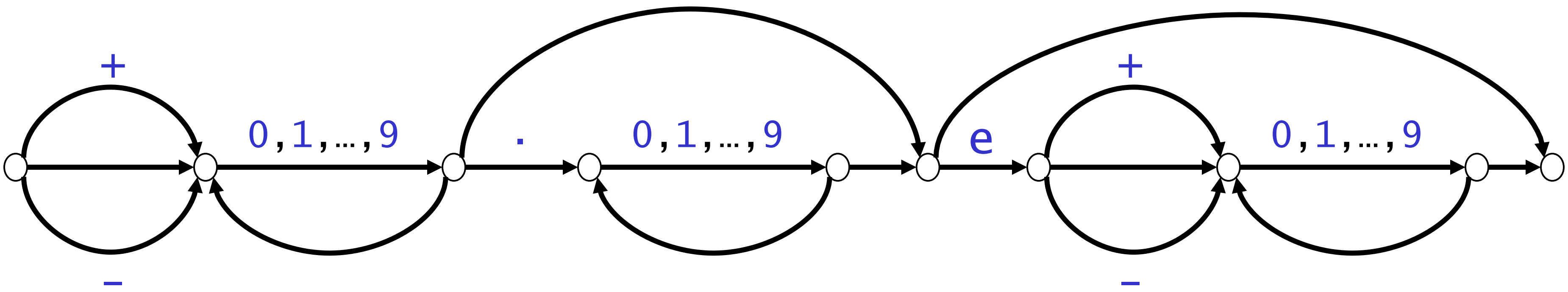
Figure 5: State Transition Table

http://teahlab.com/Moore_Finite_State_Machine_Control_Circuit/

syntax diagram

getal

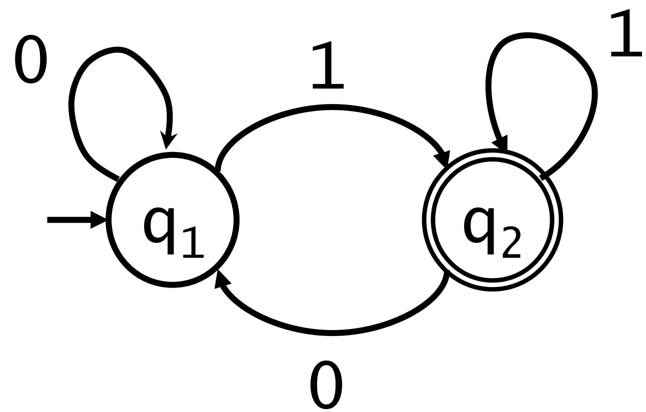




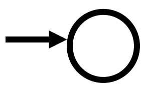
$$\{+,-,\lambda\} \cdot \{0,1,\ldots,9\}^{+} \cdot (\{\lambda\}\cup\{.\}\cdot\{0,1,\ldots,9\}^{+})$$
$$\cdot (\{\lambda\}\cup\{e\}\cdot\{+,-,\lambda\}\cdot\{0,1,\ldots,9\}^{+})$$

eindige automaat

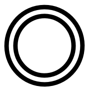
Finite automaton
Eindige automaat



$\mathcal{A}$



begintoestand



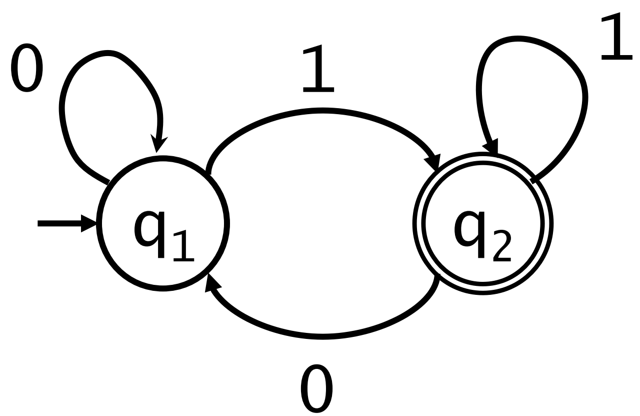
eindtoestand / accepterende toestand

Transitietabel:

	0	1
q_1	q_1	q_2
q_2	q_1	q_2

taal herkennen

Finite
automaton
Eindige
automaat



$\mathcal{A}$

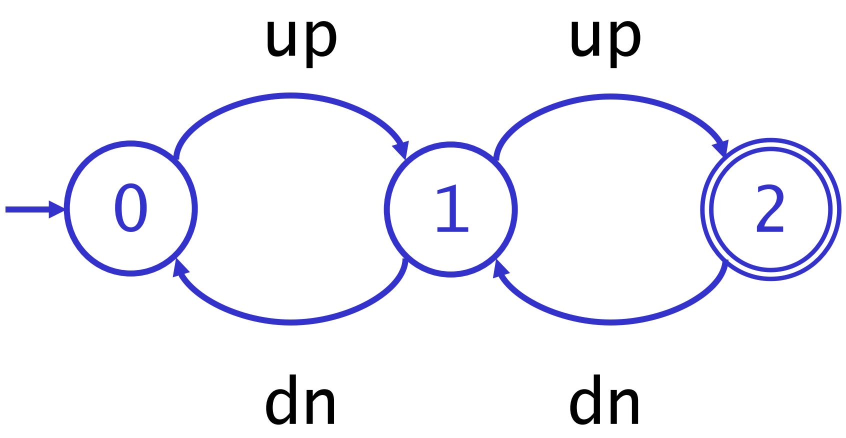
1101: $q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{1} q_2$ wel

0100: $q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{0} q_1$ niet

$L(\mathcal{A}) = \{\ w \in \{0,1\}^* \mid w \text{ eindigt op } 1\ \}$

eindige automaat

representatie als gerichte graaf
- knopen ~ toestanden
- pijlen met labels ~ acties
- begin- en eindtoestanden

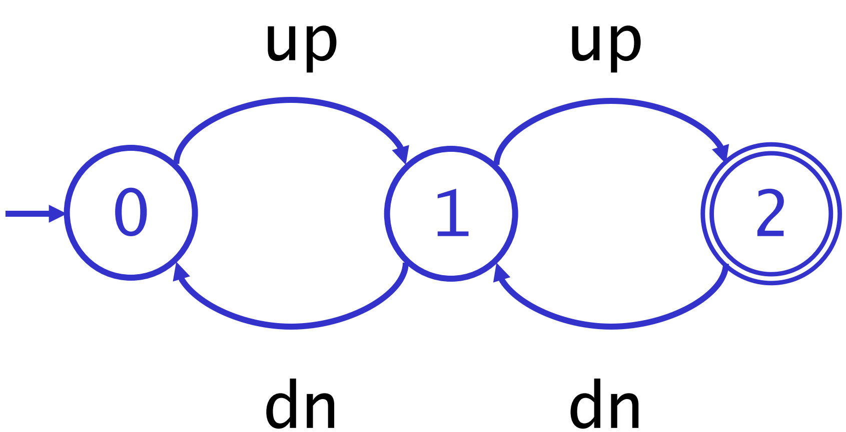


Model van een lift

definitie

een *eindige automaat* is een vijf-tupel
$\mathcal{A} = (Q,\Sigma,E,q_{in},F)$ waarbij

- Q een eindige verzameling *toestanden*
- Σ een *alfabet*
- $E \subseteq Q\times\Sigma\times Q$ verzameling *takken (pijlen)*
- $q_{in} \in Q$ de *begintoestand*
- $F \subseteq Q$ de *eindtoestanden*



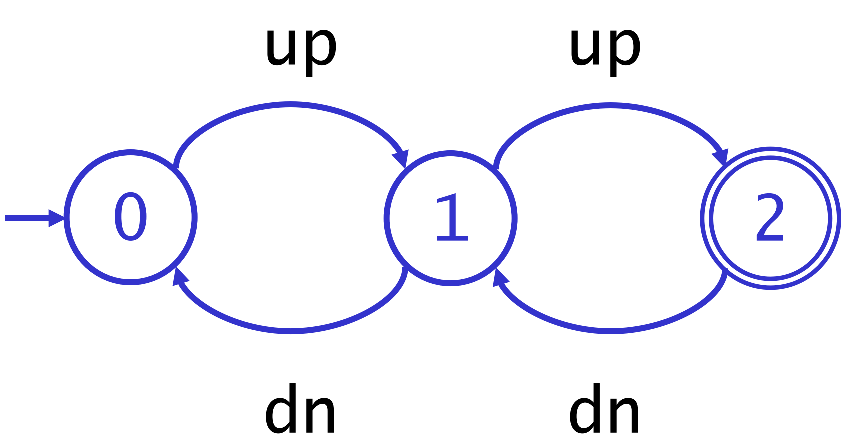
$Q = \{ 0,1,2 \}$
$\Sigma = \{ up,dn \}$
$E =$
$\{ (0,up,1), (1,up,2),$
$(2,dn,1), (1,dn,0) \}$
$q_{in} = 0$
$F = \{ 2 \}$

$\mathcal{A} = (Q,\Sigma,E,q_{in},F)$

wandeling : afwisselend knopen (toestanden) en pijlen (takken)

$p_0\ (p_0,a_0,p_1)\ p_1\ (p_1,a_2,p_2)\ \ldots\ p_{n-1}\ (p_{n-1},a_n,p_n)\ p_n$

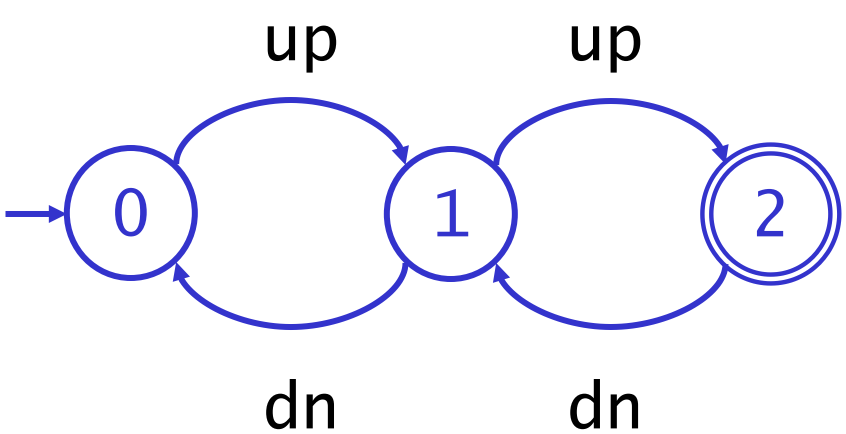
met *label* van die wandeling: $a_0 a_2 \ldots a_n$



$1 \overset{up}{\rightarrow} 2 \overset{dn}{\rightarrow} 1 \overset{dn}{\rightarrow} 0 \overset{up}{\rightarrow} 1$

gedrag: taal

$\mathcal{A} = (Q,\Sigma,E,q_{in},F)$ eindige automaat
de *taal* van $\mathcal{A}$, genoteerd $L(\mathcal{A})$, is
$\{\ x \in \Sigma^* \mid x$ is een label van een wandeling
van q_{in} naar een toestand in $F\ \}$



$\cancel{1 \xrightarrow{up} 2 \xrightarrow{dn} 1 \xrightarrow{dn} 0 \xrightarrow{up} 1}$

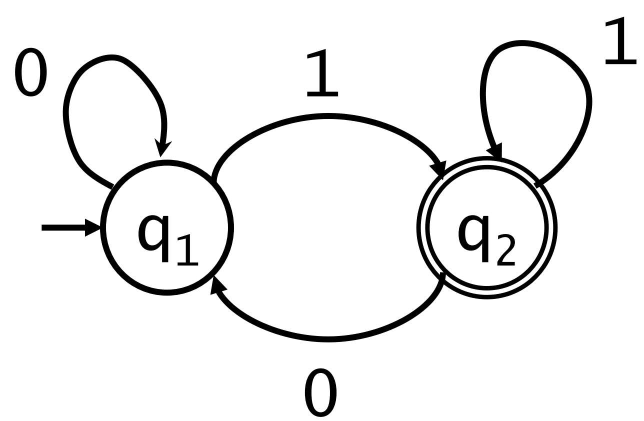
1 geen eindtoestand
geen begintoestand

$0 \xrightarrow{up} 1 \xrightarrow{dn} 0 \xrightarrow{up} 1 \xrightarrow{up} 2$

$0 \xrightarrow{up} 1 \xrightarrow{dn} 0 \xrightarrow{dn} ? \xrightarrow{up} ?$

loopt vast

reeds gezien



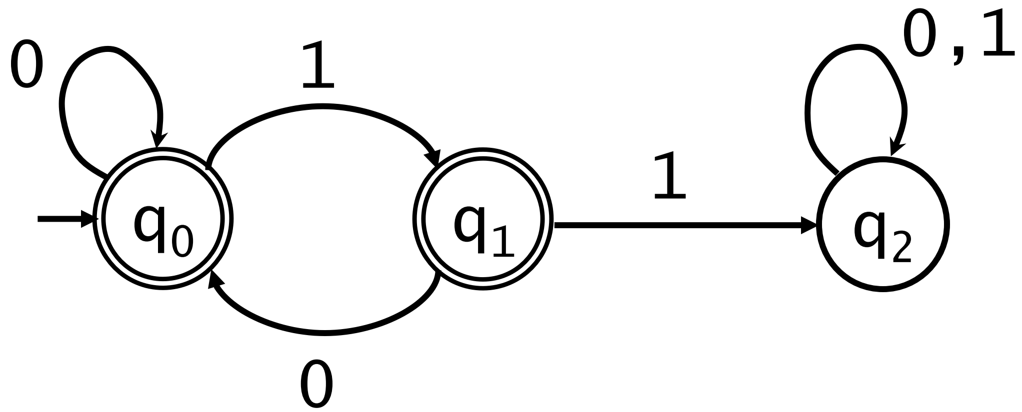
q_2 eindtoestand
/accepterende
toestand

$\mathcal{A}$

1101: $q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{1} q_2$ wel

0100: $q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_1 \xrightarrow{0} q_1$ niet

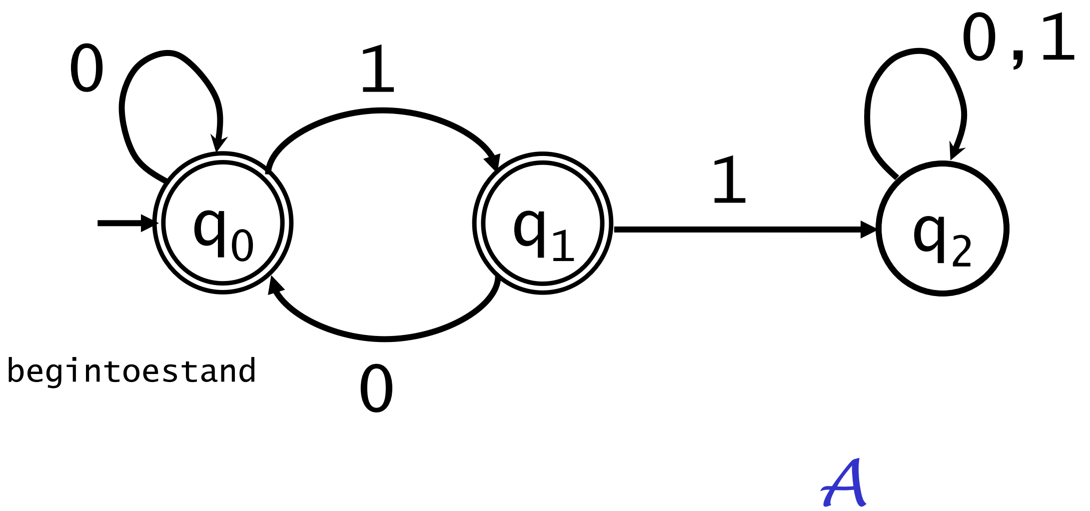
L($\mathcal{A}$) = { $w \in \{0,1\}^*$ | w eindigt op 1 }



$\mathcal{A}$

10101: wel
01101: niet
λ: wel

Wat is $L(\mathcal{A})$?



10101: wel
01101: niet
λ: wel

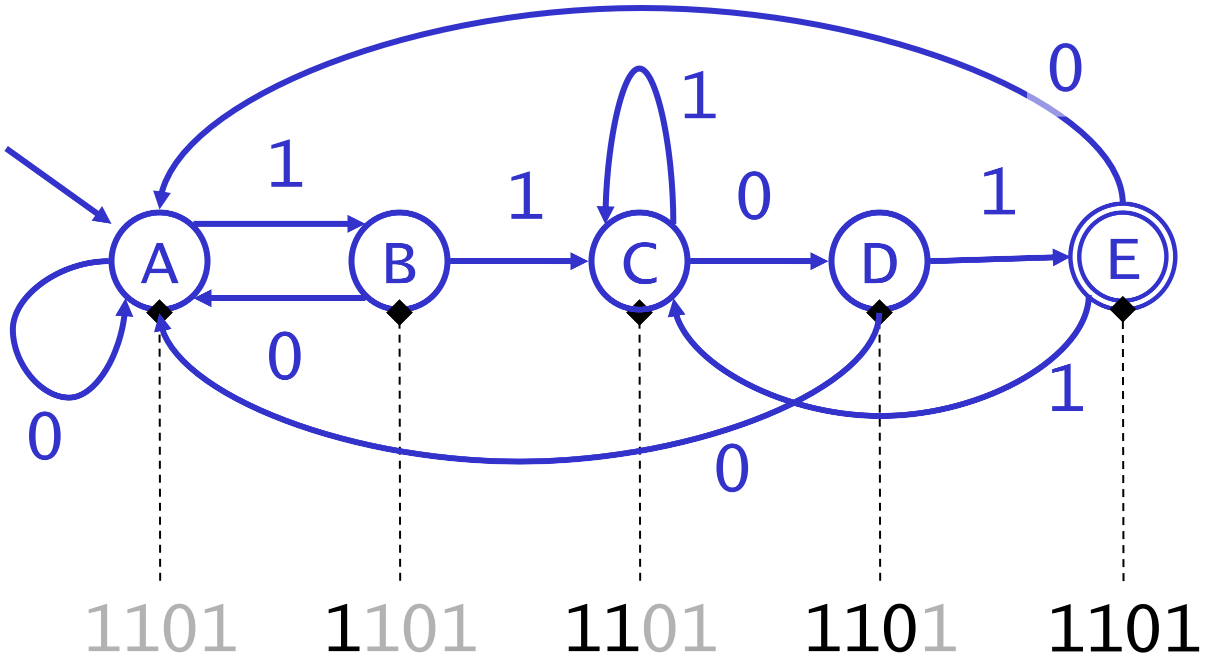
Wat is L($\mathcal{A}$)?

L($\mathcal{A}$) = { w∈{0,1}* | w bevat geen subwoord 11 }

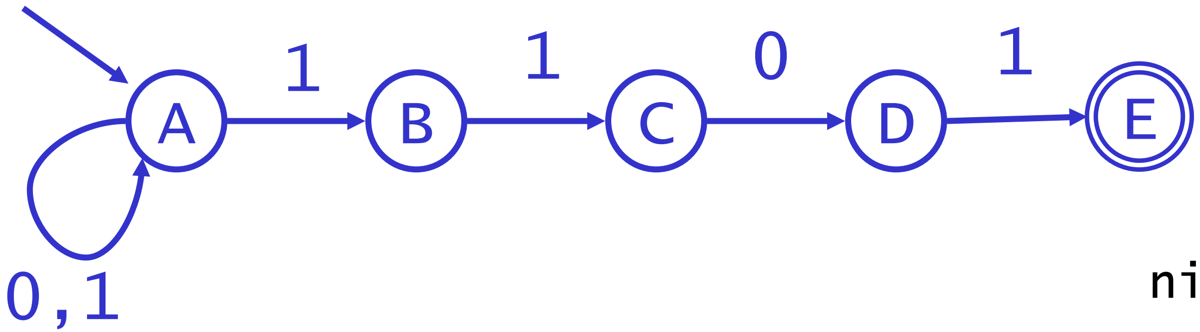
Schrijf L($\mathcal{A}$) m.b.v. vereniging, concatenatie en ster

algoritmisch vs. beschrijvend

suffix 1101



algoritme
deterministisch



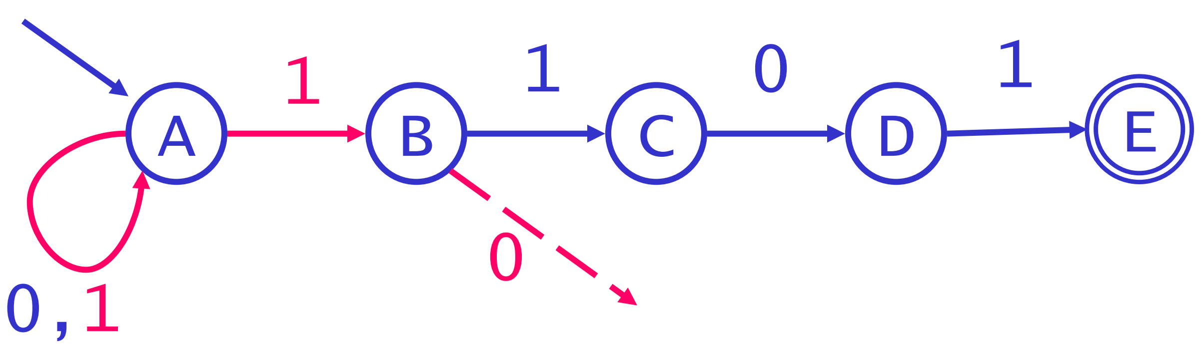
beschrijving
niet-deterministisch

algoritmisch

> 'on the fly' algoritme
> steeds van bewandeld prefix weten
  of dat zèlf tot de taal behoort
> geen keuze & niet vastlopen

definitie

een eindige automaat $\mathcal{A} = (Q,\Sigma,E,q_{in},F)$ heet *deterministisch* als voor elke toestand $p \in Q$ en elke letter $a \in \Sigma$ er precies één tak $(p,a,q) \in E$ is.



gedrag: taal

$\mathcal{A} = (Q,\Sigma,E,q_{in},F)$ eindige automaat
de *taal* van $\mathcal{A}$, genoteerd $L(\mathcal{A})$, is
$\{ x \in \Sigma^* \mid x$ is een label van een wandeling van q_{in} naar een toestand in $F \}$

als $\mathcal{A}$ deterministisch is, is de wandeling behorend bij label x uniek.

enige voorbeelden

Talen over $\{0,1\}$

1) woorden die beginnen met 0, gevolgd door nul of meer enen: $\{0\}\cdot\{1\}^*$

2) aantal 1-en is een drievoud

3) strings met substring 001

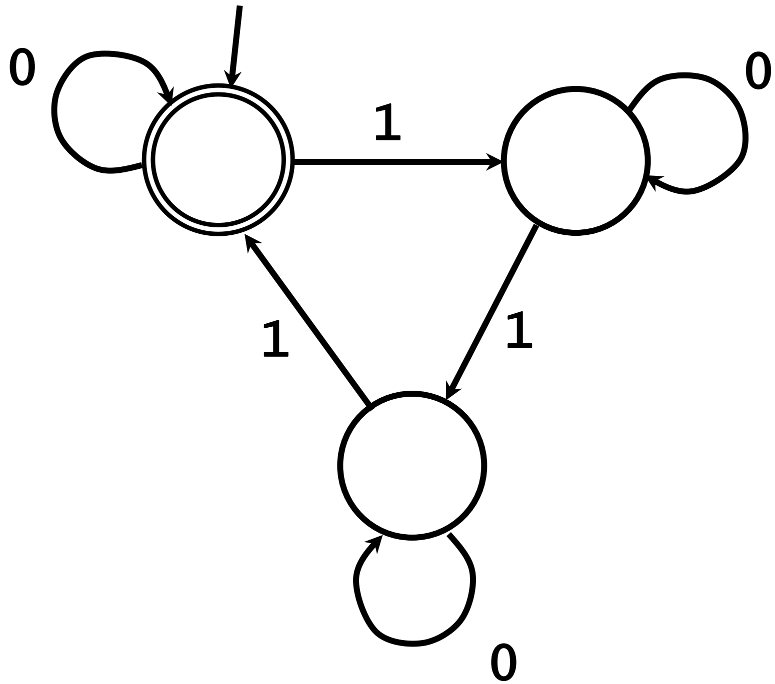
4) $\{ 01, 011, 0111 \}^*$ *tentamen fi2*

$x \in \{0,1\}^*$ $val(x) \in \mathbb{N}$: waarde als binair getal
$val(\lambda)=0$, $val(1101)=13$

5) val(x) is een drievoud

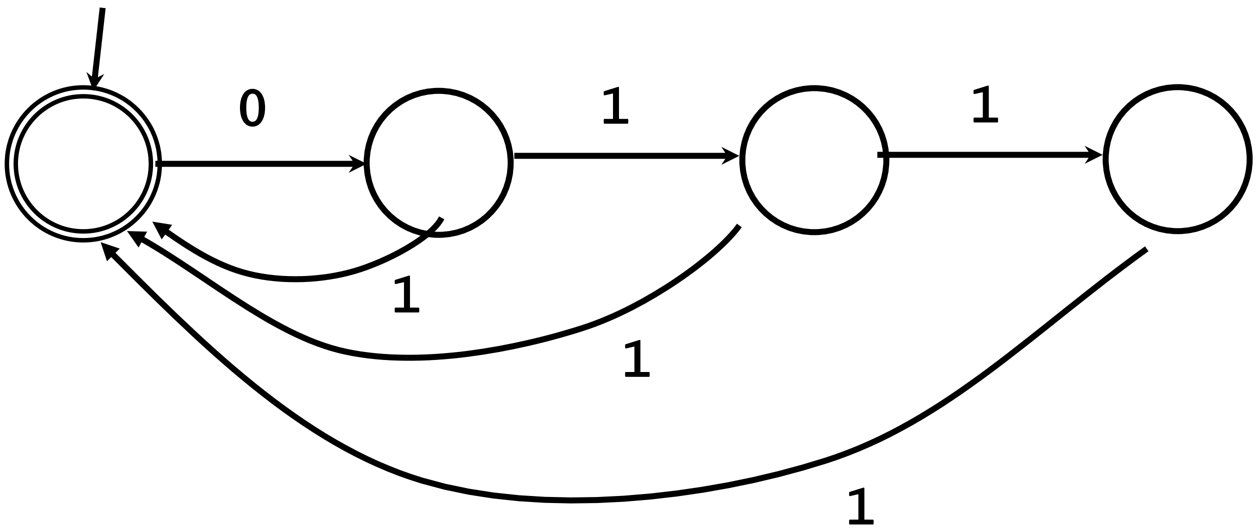
zie dictaat fi1

aantal 1-en is een drievoud



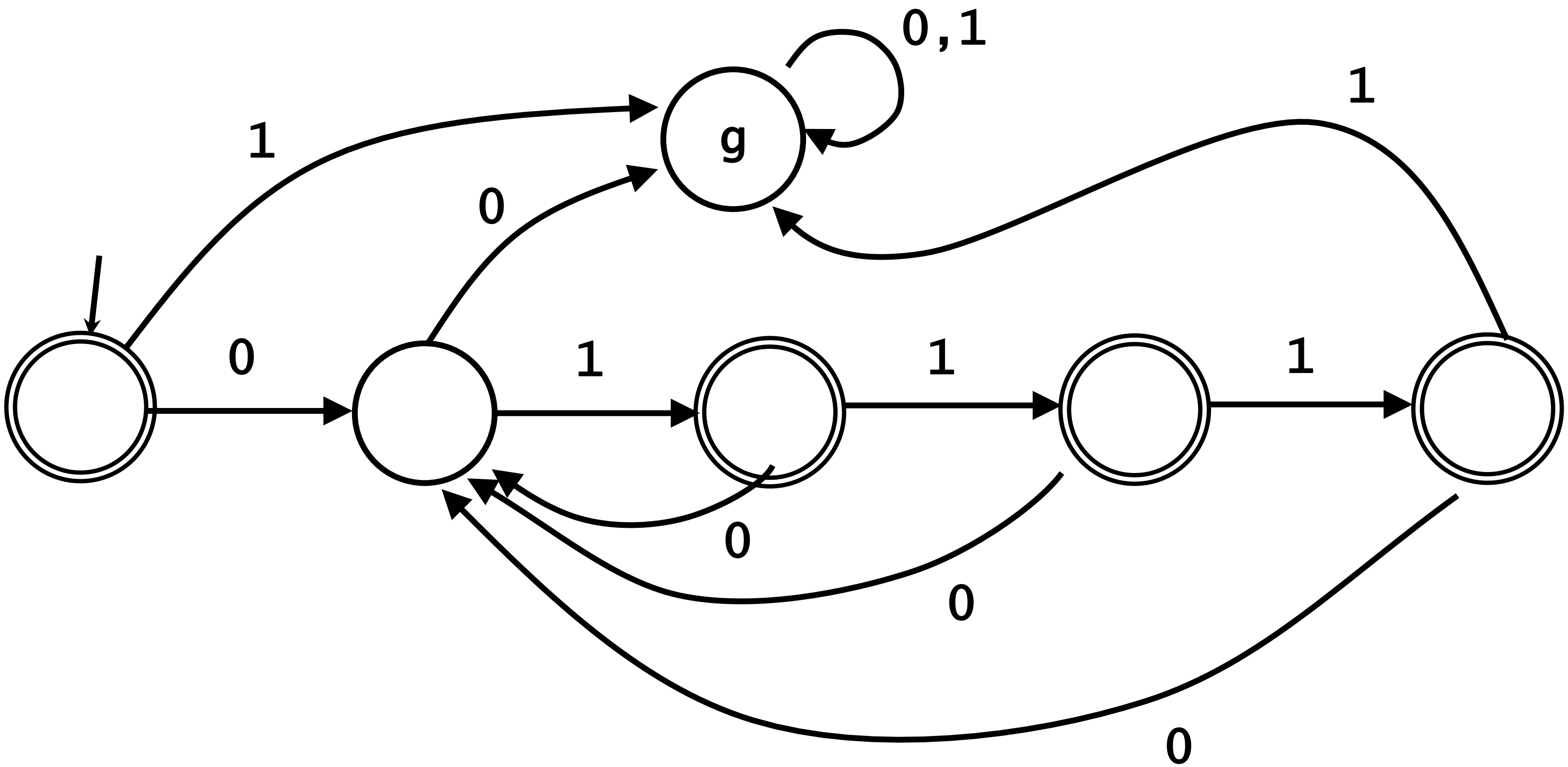
We maken die automaten op het bord; hier de uiteindelijke antwoorden van 2), 4) en 5)

{ 01, 011, 0111 }*



niet deterministisch

{ 01, 011, 0111 }*



deterministisch
inclusief 'garbage' toestand

val(w) is een drievoud

recept

toestand na lezen van $w \in \{0,1\}^*$

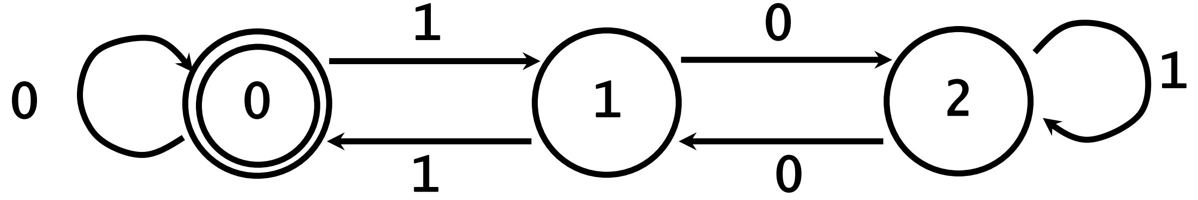
$v = val(w) \in \mathbb{N}$ modulo 3

w	$w0$	w	$w1$	(input) string $\{0,1\}^*$
v	$2v$	v	$2v+1$	waarde $\mathbb{N}$
0 1 2	0 2 1	0 1 2	1 0 2	waarde (mod 3) toestand

val(x) is een drievoud

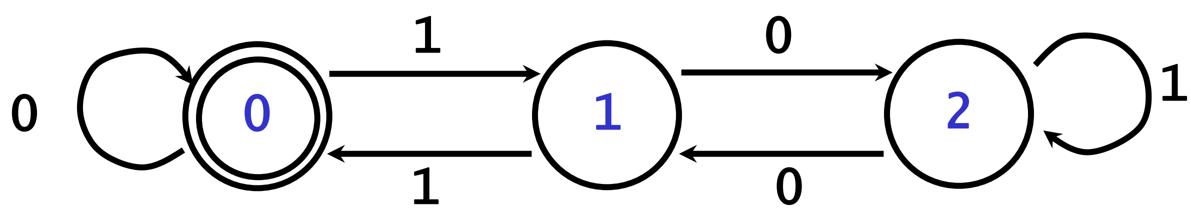
uitgewerkt

toestand na lezen van w
val(w) modulo 3



toestand ⇔ eigenschap woord

Eigenschap:
toestand na lezen van w is i ⇔
val(w) = i modulo 3
Dus in de eindtoestand geldt: x is drievoud



Basis: w = λ ok
Inductiestap: stel dat het klopt voor w

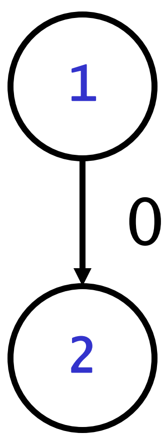
val(w)		w	↦	w0	w1	val(w0)	val(w1)
0 (3)	↔	0		0	1	2.0=0 (3)	2.0+1=1 (3)
1 (3)	↔	1		2	0	2.1=2 (3)	2.1+1=0 (3)
2 (3)	↔	2		1	2	2.2=1 (3)	2.2+1=2 (3)
	i.a.		toestand			waarde	

correctheid

formuleer een ‘**invariant**’

een relatie tussen (eigenschappen van)
woord en bereikte toestand in automaat

bewijs dat die relatie behouden blijft



⇔ w

⇔ w0

toestand	val
1	1
2	2*1=2

… inductie

meerstapsrelatie

eindige automaat $\mathcal{A} = (Q,\Sigma,E,q_{in},F)$

$E \subseteq Q \times \Sigma \times Q$ ééNstapsrelatie

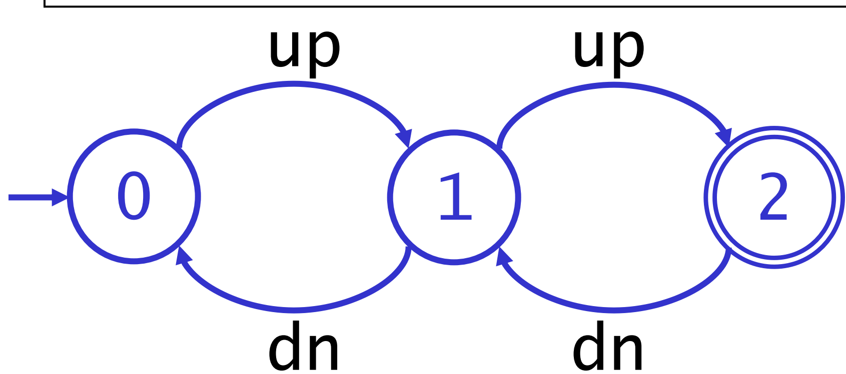
$E^* \subseteq Q \times \Sigma^* \times Q$ meerstapsrelatie:

basis $(p,\lambda,p) \in E^*$ voor elke $p \in Q$.

inductiestap

als $(p,w,q) \in E^*$ en $(q,a,r) \in E$

dan $(p,wa,r) \in E^*$



$$1 \xrightarrow{up} 2 \xrightarrow{dn} 1 \xrightarrow{dn} 0 \xrightarrow{up} 1$$

E	E*
	$(1,\lambda,1)$
(1,up,2)	(1,up,2)
(2,dn,1)	(1,up dn,1)
(1,dn,0)	(1,up dn dn,0)
(0,up,1)	(1,up dn dn up,1)

takken rijgen

basis $(p,\lambda,p) \in E^*$ voor elke $p \in Q$.

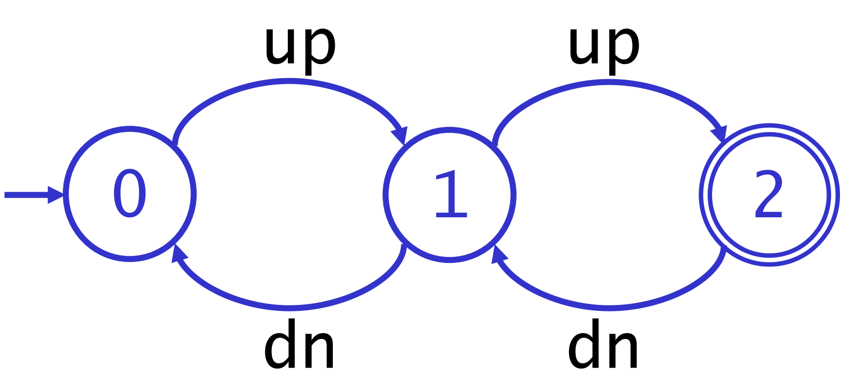
inductiestap

als $(p,w,q) \in E^*$ en $(q,a,r) \in E$

dan $(p,wa,r) \in E^*$

lemma

$(p,w,q) \in E^*$ precies als er een wandeling bestaat van p naar q met label w.



$$1 \xrightarrow{up} 2 \xrightarrow{dn} 1 \xrightarrow{dn} 0 \xrightarrow{up} 1$$

E	E*
	(1,λ,1)
(1,up,2)	(1,up,2)
(2,dn,1)	(1,up dn,1)
(1,dn,0)	(1,up dn dn,0)
(0,up,1)	(1,up dn dn up,1)

$\{\ x \in \Sigma^* \mid x$ is een label van een wandeling van q_{in} naar een toestand in $F\ \}$

Opmerking: als $\mathcal{A}$ deterministisch is, is de wandeling behorend bij label x uniek.

$\mathcal{A} = (Q,\Sigma,E,q_{in},F)$ eindige automaat

de *taal* van $\mathcal{A}$, genoteerd $L(\mathcal{A})$, is

$\{\ x \in \Sigma^* \mid (q_{in},x,q) \in E^*$ voor een toestand q in $F\ \}$

representeerbare talen

Een taal K heet *representeerbaar* als er een eindige automaat is met $L(\mathcal{A}) = K$

voorbeelden:

- $\{0\}\cdot\{1\}^*$
- drievoud aantal 1-en
- $\{01,011,0111\}^*$
- suffix 1101
- substring 001
- x met val(x) drievoud

representeerbare talen

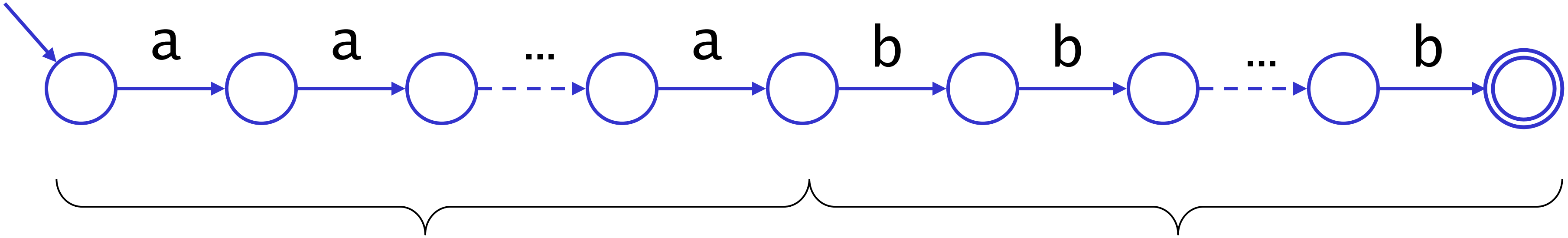
een taal K heet *representeerbaar* als er een eindige automaat is met $L(\mathcal{A}) = K$

eindig veel toestanden … niet elke taal is representeerbaar

stelling

$K = \{ a^n b^n \mid n \geq 1 \}$ is *niet* representeerbaar

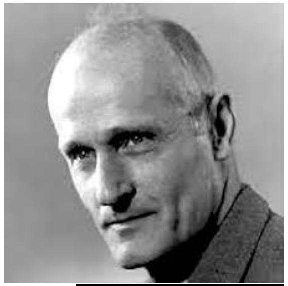
stel wél … N toestanden



N takken:
hier een toestand dubbel …

Stelling 12.2

Een taal K over een alfabet Σ is regulier dan en slechts dan als er een deterministische eindige automaat $\mathcal{A}$ bestaat zodat $K = L(\mathcal{A})$



Kleene

Opmerking: in Schaum 12.5 zijn de finite state automata per definitie deterministisch, vanwege eis (5) aldaar: er bestaat een *functie* die voor elke gegeven toestand q en letter s uit Σ *de* vervolgtoestand oplevert

(niet) deterministisch

Stelling (dictaatje H4.2)

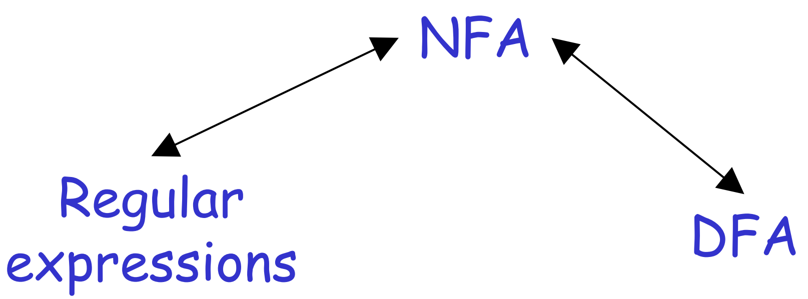
Uit een niet-deterministische eindige automaat kan een deterministische eindige automaat worden geconstrueerd voor dezelfde taal.

Zie Fundamentele informatica 2

stelling

Dezelfde familie van talen wordt gedefinieerd door

- reguliere expressies · ∪ *   ‘regulier’
- eindige automaten   ‘representeerbaar’
- deterministische eindige automaten
-



zie FI2 ‘formele talen’

chomsky hierarchie

automaten	taalfamilies	grammatica's
eindige	regulier	rechtslineair
stapel	context-vrij	context-vrij
lineair begrensd	context-gevoelig	context-gevoelig
Turing machine	recursief recursief opsombaar	type 0

zie FI2 FI3

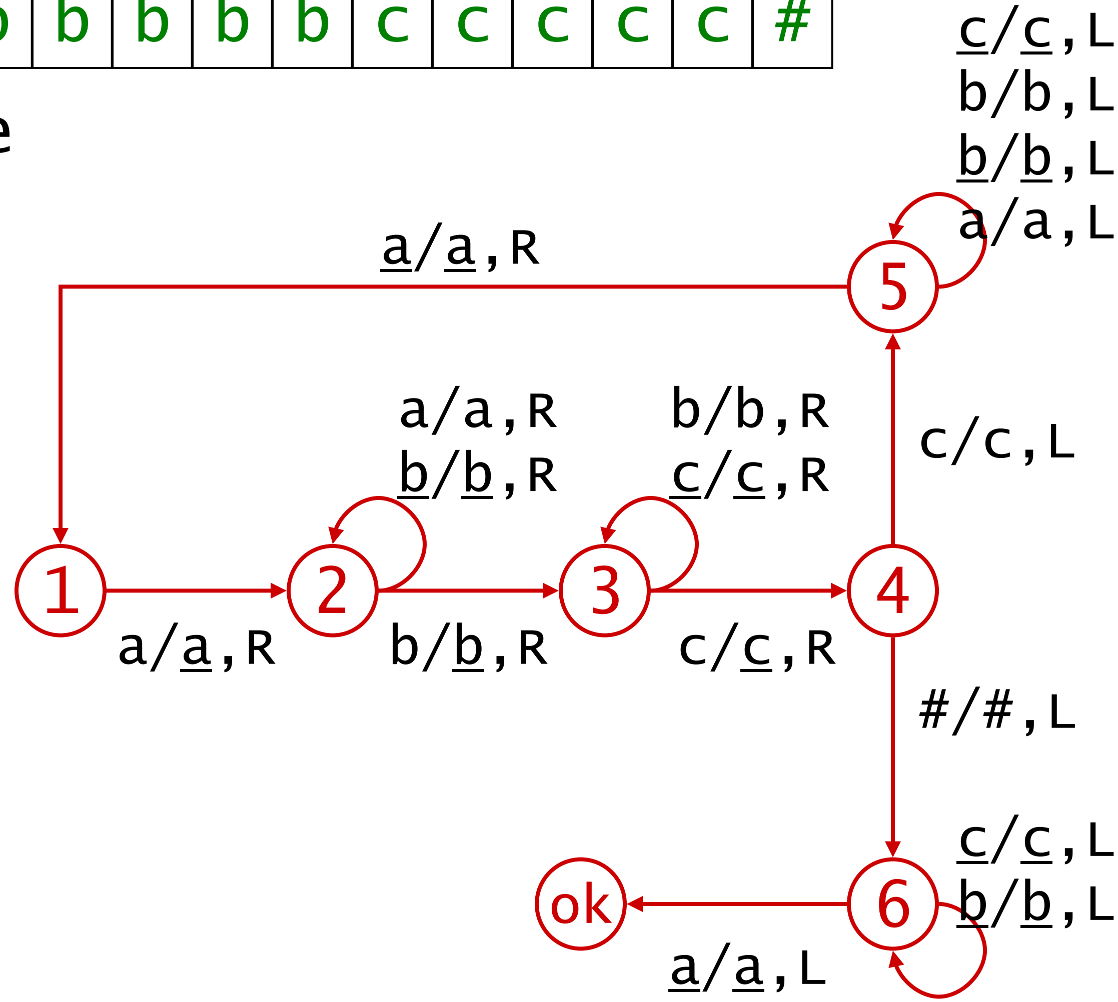
Turing machine

$\{ a^n b^n c^n \mid n \geq 0 \}$

#	a	a	a	a	a	b	b	b	b	b	c	c	c	c	c	#

tape

1. mark a
2. move to b's
   mark b
3. move to c's
   mark c
4. if another c
5. then back to a's
   goto 1.
   else back to a's
6. check marks
   stop



Laatste college volgende week:
dinsdag 11 december,
13.30 – 15.15 in zaal DS (Huygens)
hertentamen FI1 maart 2018

Werkcollege deze week
vrijdag 7 december,
9.00 – 10.45 in Snelliuszalen 402, 405

Laatste werkcollege volgende week
vrijdag 14 december,
9.00 – 10.45 in Snelliuszalen 402, 405

Vragenuur?