Alle wegen leiden naar automaten

Sam van Gool

Institute for Logic, Language and Computation
Universiteit van Amsterdam

Leve De Wiskunde! Symposium
6 april 2018

Kies je volgende vakantiebestemming

En we gaan allemaal naar...

En we gaan allemaal naar...

Verzeker je kamernummer

- Je bevindt je in een doolhof met *vier kamers*.

Verzeker je kamernummer

- Je bevindt je in een doolhof met *vier kamers*.
- Het kamernummer (**1**, **2**, **3**, of **4**) staat in onzichtbare verf op de muur geschreven, maar voor jou lijken de kamers *identiek*.

Verzeker je kamernummer

- Je bevindt je in een doolhof met *vier kamers*.
- Het kamernummer (**1**, **2**, **3**, of **4**) staat in onzichtbare verf op de muur geschreven, maar voor jou lijken de kamers *identiek*.
- In elke kamer zijn twee deuren: een *groene* en een *rode* deur.

Verzeker je kamernummer

- Je bevindt je in een doolhof met *vier kamers*.
- Het kamernummer ($\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, of $\mathbf{4}$) staat in onzichtbare verf op de muur geschreven, maar voor jou lijken de kamers *identiek*.
- In elke kamer zijn twee deuren: een *groene* en een *rode* deur.
- Elke deur leidt je, vanuit de kamer waar je bent, door een kronkelende gang, weer terug naar een kamer.

Verzeker je kamernummer

- Je bevindt je in een doolhof met *vier kamers*.
- Het kamernummer (**1**, **2**, **3**, of **4**) staat in onzichtbare verf op de muur geschreven, maar voor jou lijken de kamers *identiek*.
- In elke kamer zijn twee deuren: een *groene* en een *rode* deur.
- Elke deur leidt je, vanuit de kamer waar je bent, door een kronkelende gang, weer terug naar een kamer.
- Je hebt *geen idee* in welke kamer je begint.

Verzeker je kamernummer

- Je bevindt je in een doolhof met *vier kamers*.
- Het kamernummer ($\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, of $\mathbf{4}$) staat in onzichtbare verf op de muur geschreven, maar voor jou lijken de kamers *identiek*.
- In elke kamer zijn twee deuren: een *groene* en een *rode* deur.
- Elke deur leidt je, vanuit de kamer waar je bent, door een kronkelende gang, weer terug naar een kamer.
- Je hebt *geen idee* in welke kamer je begint.
- Je mag *negen keer* een deur doorgaan.

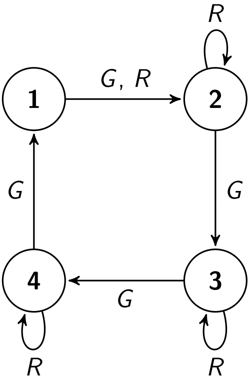
Verzeker je kamernummer

- Je bevindt je in een doolhof met *vier kamers*.
- Het kamernummer (**1**, **2**, **3**, of **4**) staat in onzichtbare verf op de muur geschreven, maar voor jou lijken de kamers *identiek*.
- In elke kamer zijn twee deuren: een *groene* en een *rode* deur.
- Elke deur leidt je, vanuit de kamer waar je bent, door een kronkelende gang, weer terug naar een kamer.
- Je hebt *geen idee* in welke kamer je begint.
- Je mag *negen keer* een deur doorgaan.
- Als je na die negen keer weet in *welke kamer* je je bevindt, win je een pot met goud.

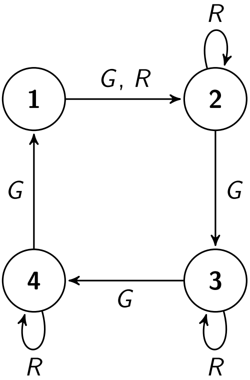
Verzeker je kamernummer

- Je bevindt je in een doolhof met *vier kamers*.
- Het kamernummer ($\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, of $\mathbf{4}$) staat in onzichtbare verf op de muur geschreven, maar voor jou lijken de kamers *identiek*.
- In elke kamer zijn twee deuren: een *groene* en een *rode* deur.
- Elke deur leidt je, vanuit de kamer waar je bent, door een kronkelende gang, weer terug naar een kamer.
- Je hebt *geen idee* in welke kamer je begint.
- Je mag *negen keer* een deur doorgaan.
- Als je na die negen keer weet in *welke kamer* je je bevindt, win je een pot met goud.
- Je krijgt ook een *kaart van het doolhof*!

De kaart van het doolhof

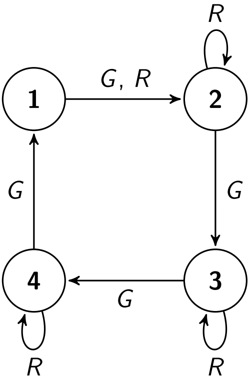


De kaart van het doolhof



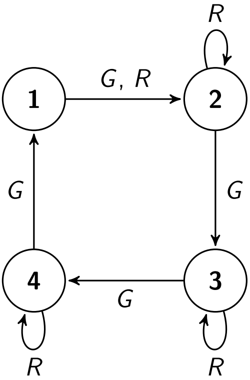
- Elke *groene* deur brengt je één kamer 'verder':
  $G : \mathbf{1} \to \mathbf{2},\ \mathbf{2} \to \mathbf{3},\ \mathbf{3} \to \mathbf{4},\ \mathbf{4} \to \mathbf{1}.$

De kaart van het doolhof



- Elke *groene* deur brengt je één kamer ‘verder’:
  G : **1** → **2**, **2** → **3**, **3** → **4**, **4** → **1**.
- De *rode* deur in kamer **1** brengt je naar kamer **2**, maar in kamers **2** - **4** kom je terug in dezelfde kamer:
  R : **1** → **2**, **2** → **2**, **3** → **3**, **4** → **4**.

De kaart van het doolhof



- Elke *groene* deur brengt je één kamer 'verder':
  G : **1** $\to$ **2**, **2** $\to$ **3**, **3** $\to$ **4**, **4** $\to$ **1**.
- De *rode* deur in kamer **1** brengt je naar kamer **2**, maar in kamers **2** - **4** kom je terug in dezelfde kamer:
  R : **1** $\to$ **2**, **2** $\to$ **2**, **3** $\to$ **3**, **4** $\to$ **4**.

Doel: na (hoogstens) negen deuren zeker weten in welke kamer je bent, ongeacht waar je begint.

Geldige e-mailadressen herkennen

- Een webformulier vraagt de gebruiker om een *e-mailadres*.

Geldige e-mailadressen herkennen

- Een webformulier vraagt de gebruiker om een *e-mailadres*.
- Een e-mailadres is *geldig* dan en slechts dan als:

Geldige e-mailadressen herkennen

- Een webformulier vraagt de gebruiker om een *e-mailadres*.
- Een e-mailadres is *geldig* dan en slechts dan als:
  - het eerst minstens één alfanumeriek teken (t) bevat, en mogelijk meer tekens (t) en punten (.);

Geldige e-mailadressen herkennen

- Een webformulier vraagt de gebruiker om een *e-mailadres*.
- Een e-mailadres is *geldig* dan en slechts dan als:
  - het eerst minstens één alfanumeriek teken (t) bevat, en mogelijk meer tekens (t) en punten (.);
  - daarna op een gegeven moment een apenstaartje (@);

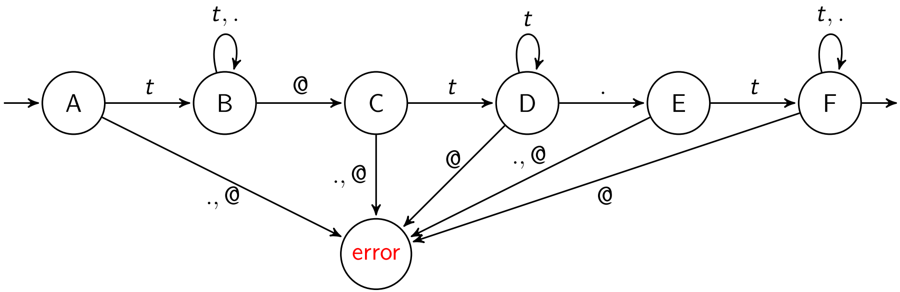
Geldige e-mailadressen herkennen

- Een webformulier vraagt de gebruiker om een *e-mailadres*.
- Een e-mailadres is *geldig* dan en slechts dan als:
  - het eerst minstens één alfanumeriek teken (t) bevat, en mogelijk meer tekens (t) en punten (.);
  - daarna op een gegeven moment een apenstaartje (@);
  - daarna minstens één alfanumeriek teken (t), gevolgd door een punt (.), een teken (t), en daarna mogelijk nog een aantal tekens en/of punten.

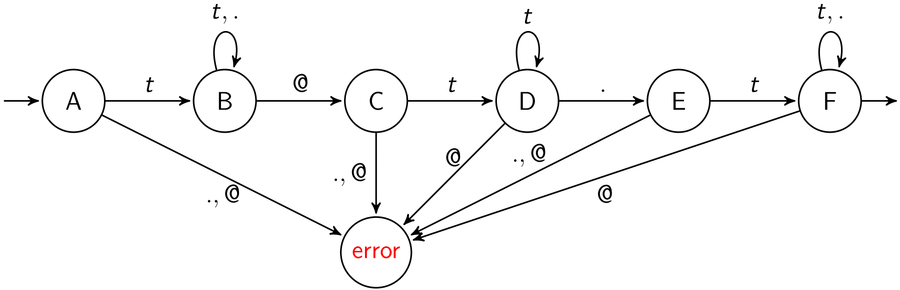
Geldige e-mailadressen herkennen

- Een webformulier vraagt de gebruiker om een *e-mailadres*.
- Een e-mailadres is *geldig* dan en slechts dan als:
  - het eerst minstens één alfanumeriek teken (t) bevat, en mogelijk meer tekens (t) en punten (.);
  - daarna op een gegeven moment een apenstaartje (`@`);
  - daarna minstens één alfanumeriek teken (t), gevolgd door een punt (.), een teken (t), en daarna mogelijk nog een aantal tekens en/of punten.
- Hoe kun je *automatisch herkennen* of een e-mailadres geldig is?

Automatisch schema voor geldige e-mailadressen

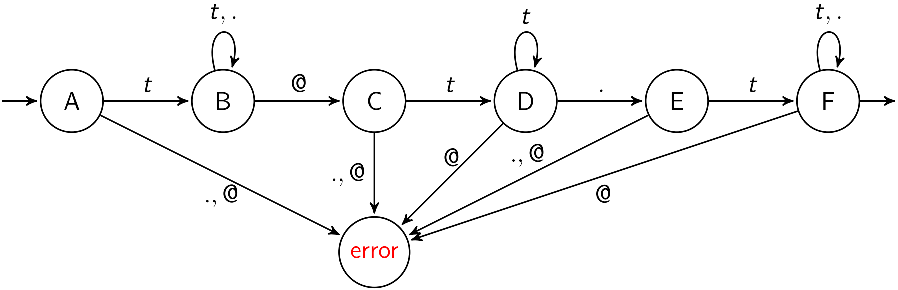


Automatisch schema voor geldige e-mailadressen



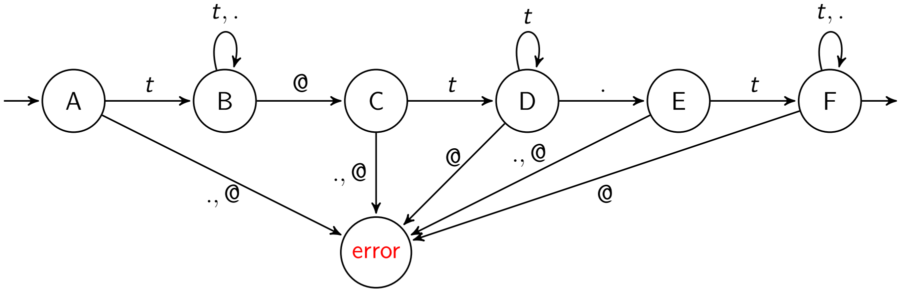
Als we 'S.J.vanGool@uva.nl' invoeren bij A, dan komen we uit in

Automatisch schema voor geldige e-mailadressen



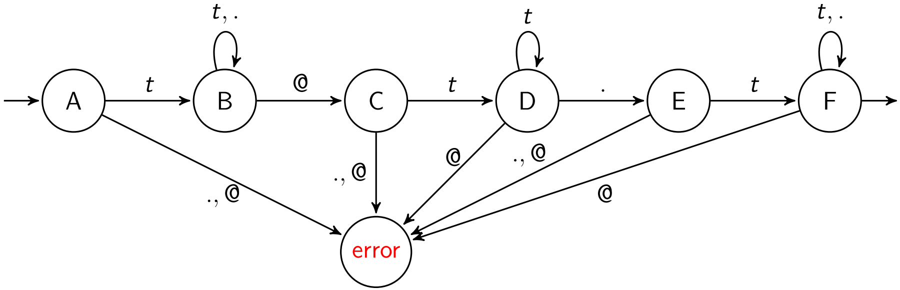
Als we 'S.J.vanGool@uva.nl' invoeren bij A, dan komen we uit in F.

Automatisch schema voor geldige e-mailadressen



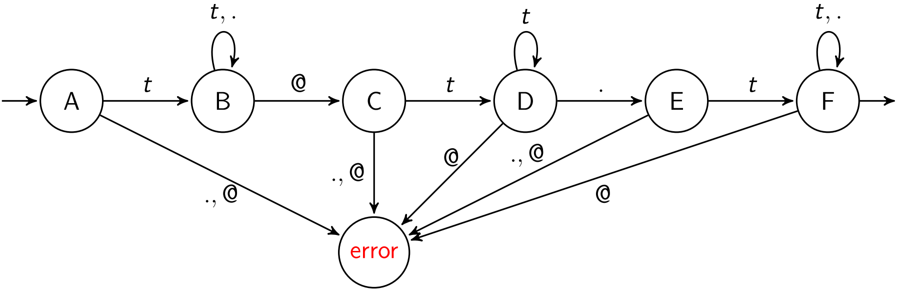
Als we ‘S.J.vanGool@uva.nl’ invoeren bij A, dan komen we uit in F.
Als we ‘sam@van@gool’ invoeren bij A, dan komen we uit in

Automatisch schema voor geldige e-mailadressen



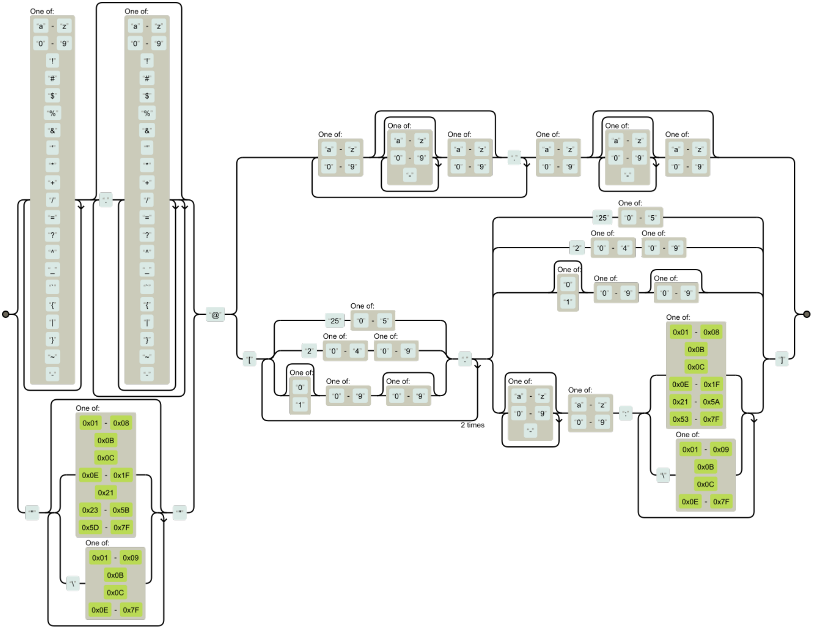
Als we ‘S.J.vanGool@uva.nl’ invoeren bij A, dan komen we uit in F.
Als we ‘sam@van@gool’ invoeren bij A, dan komen we uit in error.

Automatisch schema voor geldige e-mailadressen



Als we 'S.J.vanGool@uva.nl' invoeren bij A, dan komen we uit in F.
Als we 'sam@van@gool' invoeren bij A, dan komen we uit in error.
Dit is een *versimpelde* versie van de werkelijkheid...

Werkelijk automatisch schema voor e-mailadressen



Wat is een automaat?

Een automaat bestaat uit:

- een eindig aantal *toestanden* $q_1, \ldots, q_n$,

Wat is een automaat?

Een automaat bestaat uit:

- een eindig aantal *toestanden* $q_1, \ldots, q_n$,
- een eindig aantal *letters* $a_1, \ldots, a_m$,

Wat is een automaat?

Een automaat bestaat uit:

- een eindig aantal *toestanden* $q_1, \ldots, q_n$,
- een eindig aantal *letters* $a_1, \ldots, a_m$,
- een *transitie-functie* die aan elke toestand q en letter a een nieuwe toestand, $\delta_a(q)$, toekent.

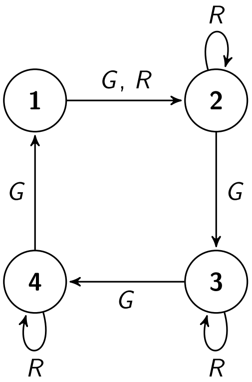
Wat is een automaat?

Een automaat bestaat uit:

- een eindig aantal *toestanden* $q_1, \ldots, q_n$,
- een eindig aantal *letters* $a_1, \ldots, a_m$,
- een *transitie-functie* die aan elke toestand q en letter a een nieuwe toestand, $\delta_a(q)$, toekent.

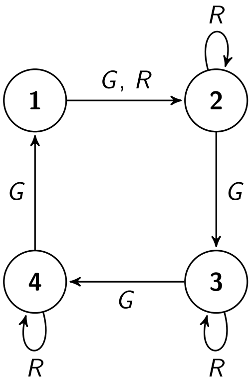
In de laatste toepassing gaven we verder ook een *begin-toestand* en *accepterende toestanden* aan.

Het doolhof als automaat



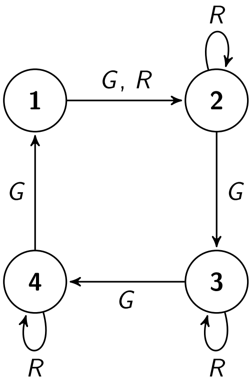
- de *toestanden*

Het doolhof als automaat



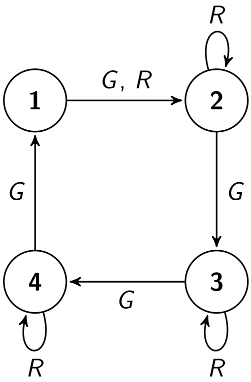
- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;

Het doolhof als automaat



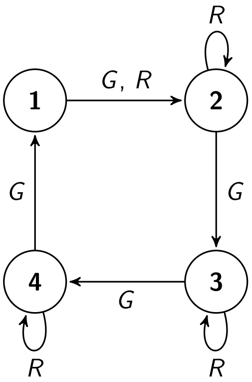
- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters*

Het doolhof als automaat



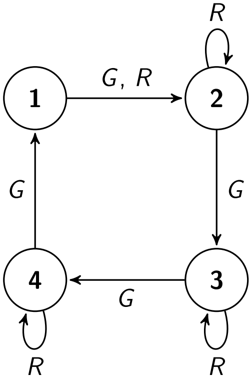
- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;

Het doolhof als automaat



- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

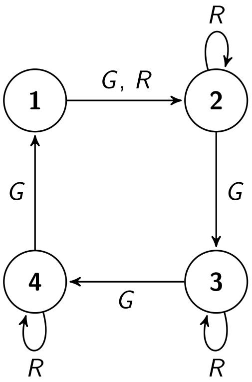
Het doolhof als automaat



- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	
1	**2**	**2**	
2	**3**	**2**	
3	**4**	**3**	
4	**1**	**4**	

Het doolhof als automaat

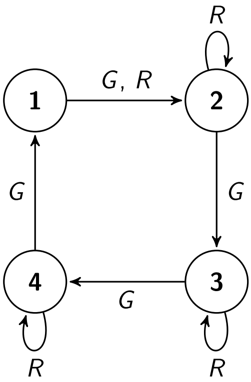


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	
1	**2**	**2**	
2	**3**	**2**	
3	**4**	**3**	
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.

Het doolhof als automaat

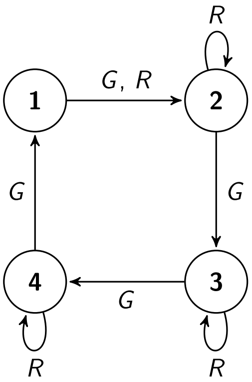


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	
1	**2**	**2**	
2	**3**	**2**	
3	**4**	**3**	
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft:

Het doolhof als automaat

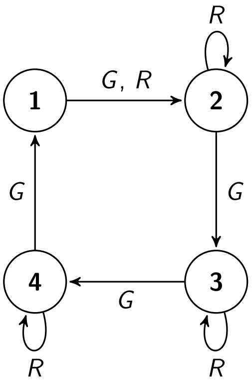


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	
2	**3**	**2**	
3	**4**	**3**	
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: $\mathbf{1} \to$

Het doolhof als automaat

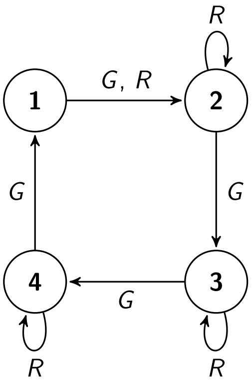


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	**1**
2	**3**	**2**	
3	**4**	**3**	
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: $\mathbf{1} \to \mathbf{1}$,

Het doolhof als automaat

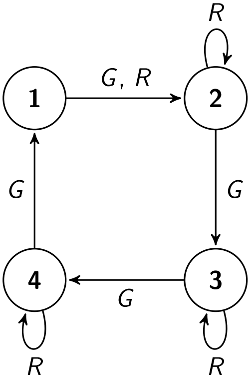


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	**1**
2	**3**	**2**	
3	**4**	**3**	
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: $\mathbf{1} \to \mathbf{1}$, $\mathbf{2} \to$

Het doolhof als automaat

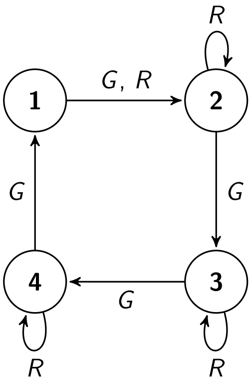


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	**1**
2	**3**	**2**	**1**
3	**4**	**3**	
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: $\mathbf{1} \to \mathbf{1}$, $\mathbf{2} \to \mathbf{1}$,

Het doolhof als automaat

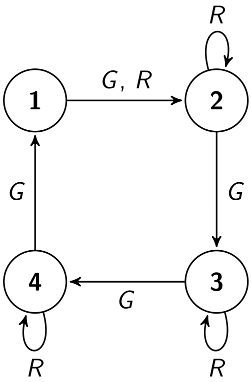


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	**1**
2	**3**	**2**	**1**
3	**4**	**3**	
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: **1** $\to$ **1**, **2** $\to$ **1**, **3** $\to$

Het doolhof als automaat

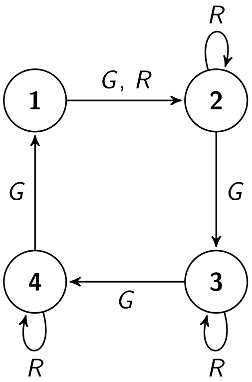


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	**1**
2	**3**	**2**	**1**
3	**4**	**3**	**2**
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: $\mathbf{1} \to \mathbf{1}$, $\mathbf{2} \to \mathbf{1}$, $\mathbf{3} \to \mathbf{2}$,

Het doolhof als automaat

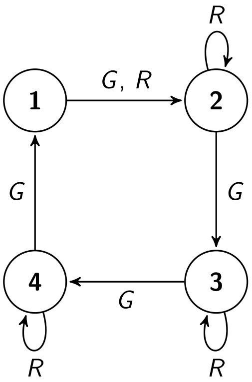


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	**1**
2	**3**	**2**	**1**
3	**4**	**3**	**2**
4	**1**	**4**	

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: **1** $\to$ **1**, **2** $\to$ **1**, **3** $\to$ **2**, **4** $\to$

Het doolhof als automaat

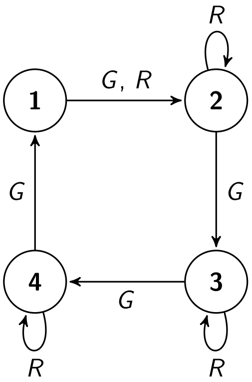


- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	**1**
2	**3**	**2**	**1**
3	**4**	**3**	**2**
4	**1**	**4**	**3**

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: **1** $\rightarrow$ **1**, **2** $\rightarrow$ **1**, **3** $\rightarrow$ **2**, **4** $\rightarrow$ **3**.

Het doolhof als automaat



- de *toestanden* zijn de kamers, **1**, **2**, **3**, **4**;
- de *letters* zijn de kleuren, G en R;
- de gangen geven de *transitie-functie*:

q	$\delta_G(q)$	$\delta_R(q)$	$\delta_{RGGG}(q)$
1	**2**	**2**	**1**
2	**3**	**2**	**1**
3	**4**	**3**	**2**
4	**1**	**4**	**3**

Ieder *rijtje letters* geeft ook een transitie-functie.
Bijvoorbeeld, $RGGG$ geeft: **1** $\to$ **1**, **2** $\to$ **1**, **3** $\to$ **2**, **4** $\to$ **3**.

"Een automaat is een werking door de vrije halfgroep over de letters op de verzameling toestanden."

Synchroniserende woorden

- Een eindig rijtje letters noemen we een *woord*.

Synchroniserende woorden

- Een eindig rijtje letters noemen we een *woord*.
- We noemen een woord *synchroniserend* voor een automaat als de automaat zich na het lezen van het woord altijd in dezelfde eindtoestand bevindt, ongeacht in welke toestand hij begint.

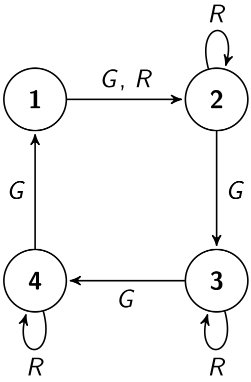
Synchroniserende woorden

- Een eindig rijtje letters noemen we een *woord*.
- We noemen een woord *synchroniserend* voor een automaat als de automaat zich na het lezen van het woord altijd in dezelfde eindtoestand bevindt, ongeacht in welke toestand hij begint.
- Korter gezegd: een woord w heet synchroniserend als $\delta_w(q) = \delta_w(r)$ voor alle toestanden q, r van de automaat.

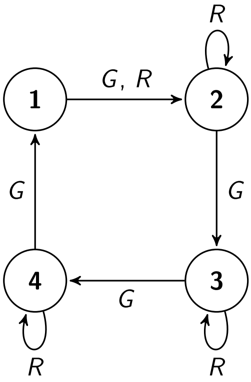
Synchroniserende woorden

- Een eindig rijtje letters noemen we een *woord*.
- We noemen een woord *synchroniserend* voor een automaat als de automaat zich na het lezen van het woord altijd in dezelfde eindtoestand bevindt, ongeacht in welke toestand hij begint.
- Korter gezegd: een woord w heet synchroniserend als $\delta_w(q) = \delta_w(r)$ voor alle toestanden q, r van de automaat.
- Een automaat heet *synchroniseerbaar* als er een synchroniserend woord voor de automaat bestaat.

Voorbeeld van een synchroniserend woord voor het doolhof

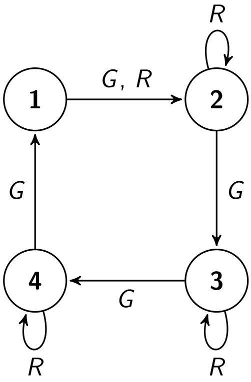


Voorbeeld van een synchroniserend woord voor het doolhof



- Bekijk het woord $w = RGGRGRGGGR$.

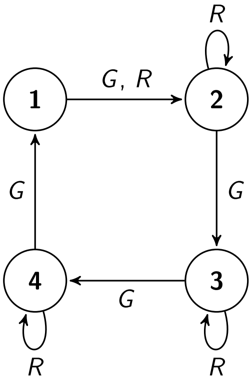
Voorbeeld van een synchroniserend woord voor het doolhof



- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	

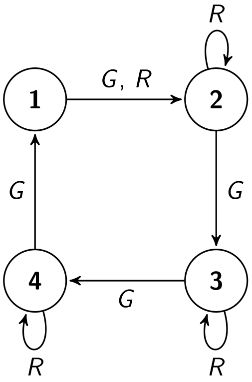
Voorbeeld van een synchroniserend woord voor het doolhof



- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	

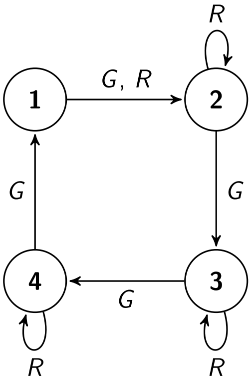
Voorbeeld van een synchroniserend woord voor het doolhof



- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	**2**
3	

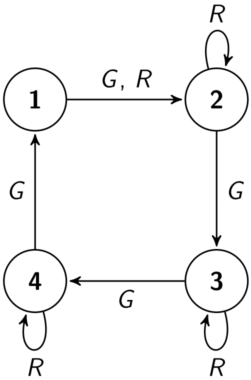
Voorbeeld van een synchroniserend woord voor het doolhof



- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	**2**
3	**2**
4	

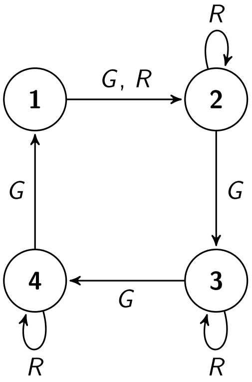
Voorbeeld van een synchroniserend woord voor het doolhof



- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	**2**
3	**2**
4	**2**

Voorbeeld van een synchroniserend woord voor het doolhof

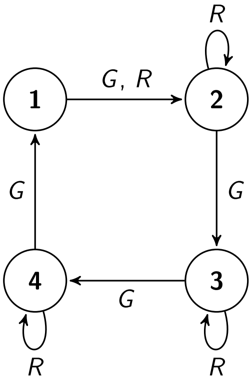


- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	**2**
3	**2**
4	**2**

- Het woord w is synchroniserend voor de doolhof-automaat.

Voorbeeld van een synchroniserend woord voor het doolhof



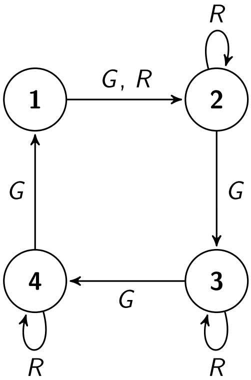
- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	**2**
3	**2**
4	**2**

- Het woord w is synchroniserend voor de doolhof-automaat.

- Het woord w heeft lengte

Voorbeeld van een synchroniserend woord voor het doolhof



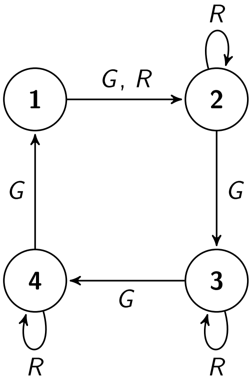
- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	**2**
3	**2**
4	**2**

- Het woord w is synchroniserend voor de doolhof-automaat.

- Het woord w heeft lengte **10**.

Voorbeeld van een synchroniserend woord voor het doolhof



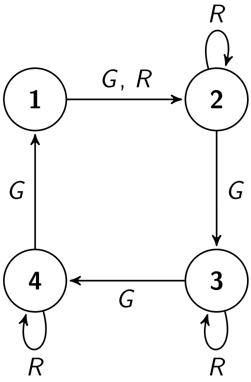
- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	**2**
3	**2**
4	**2**

- Het woord w is synchroniserend voor de doolhof-automaat.

- Het woord w heeft lengte **10**.
- Je kunt dit woord dus *niet* gebruiken om het goud te winnen.

Voorbeeld van een synchroniserend woord voor het doolhof



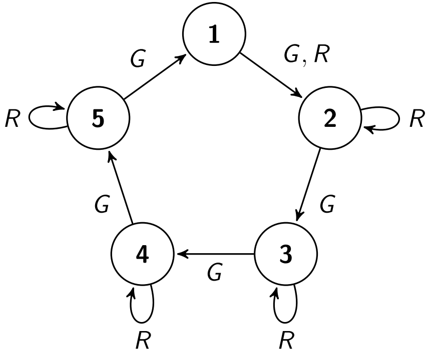
- Bekijk het woord $w = RGGRGRGGGR$.

q	$\delta_w(q)$
1	**2**
2	**2**
3	**2**
4	**2**

- Het woord w is synchroniserend voor de doolhof-automaat.

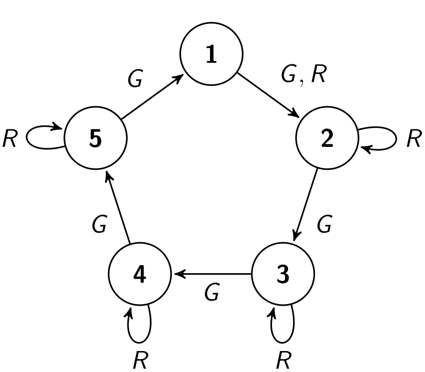
- Het woord w heeft lengte **10**.
- Je kunt dit woord dus *niet* gebruiken om het goud te winnen.
- De uitdaging blijft: vind een synchroniserend woord van **9** letters.

Meer doolhoven, meer uitdagingen

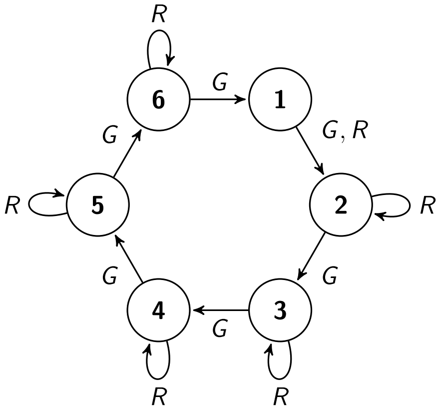


Een synchroniserend woord van lengte **16**?

Meer doolhoven, meer uitdagingen



Een synchroniserend woord van lengte **16**?



Een synchroniserend woord van lengte **25**?

Korte synchroniserende woorden

Het vermoeden van Černý (1964)

Als een synchroniseerbare automaat n toestanden heeft, dan bestaat er altijd een synchroniserend woord van lengte hoogstens $(n-1)^2$.

Korte synchroniserende woorden

Het vermoeden van Černý (1964)

Als een synchroniseerbare automaat n toestanden heeft, dan bestaat er altijd een synchroniserend woord van lengte hoogstens $(n-1)^2$.

- Černý vond de doolhof-automaten hierboven, waarin het *kortste* synchroniserende woord steeds lengte $(n-1)^2$ heeft.

Korte synchroniserende woorden

Het vermoeden van Černý (1964)

Als een synchroniseerbare automaat n toestanden heeft, dan bestaat er altijd een synchroniserend woord van lengte hoogstens $(n-1)^2$.

- Černý vond de doolhof-automaten hierboven, waarin het *kortste* synchroniserende woord steeds lengte $(n-1)^2$ heeft.
- We schrijven $C(n)$ voor de maximale lengte van het kortste synchroniserende woord voor een synchroniseerbare automaat met n toestanden.

Korte synchroniserende woorden

Het vermoeden van Černý (1964)

Als een synchroniseerbare automaat n toestanden heeft, dan bestaat er altijd een synchroniserend woord van lengte hoogstens $(n-1)^2$.

- Černý vond de doolhof-automaten hierboven, waarin het *kortste* synchroniserende woord steeds lengte $(n-1)^2$ heeft.
- We schrijven $C(n)$ voor de maximale lengte van het kortste synchroniserende woord voor een synchroniseerbare automaat met n toestanden.

Stelling van Frankl en Pin (1983)

Voor iedere n geldt: $C(n) \leq \dfrac{n^3 - n}{6}$.

Korte synchroniserende woorden

Het vermoeden van Černý (1964)

Als een synchroniseerbare automaat n toestanden heeft, dan bestaat er altijd een synchroniserend woord van lengte hoogstens $(n-1)^2$.

- Černý vond de doolhof-automaten hierboven, waarin het *kortste* synchroniserende woord steeds lengte $(n-1)^2$ heeft.
- We schrijven $C(n)$ voor de maximale lengte van het kortste synchroniserende woord voor een synchroniseerbare automaat met n toestanden.

Stelling van Frankl en Pin (1983)

Voor iedere n geldt: $C(n) \leq \dfrac{n^3 - n}{6}$.

- Het bewijs is niet makkelijk. Op de handout: $C(n) \leq \frac{n^3-2n^2+n}{2}$.

Korte synchroniserende woorden

Het vermoeden van Černý (1964)

Als een synchroniseerbare automaat n toestanden heeft, dan bestaat er altijd een synchroniserend woord van lengte hoogstens $(n-1)^2$.

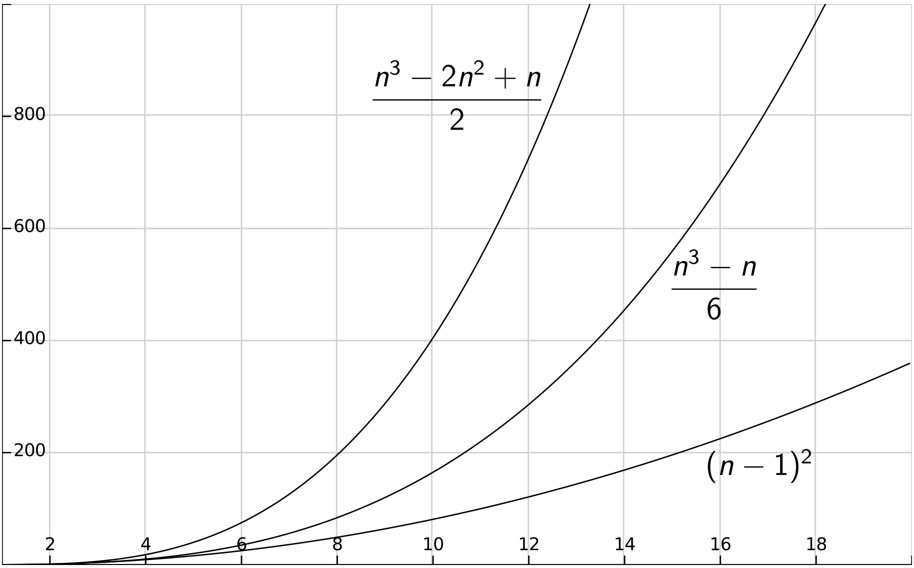
- Černý vond de doolhof-automaten hierboven, waarin het *kortste* synchroniserende woord steeds lengte $(n-1)^2$ heeft.
- We schrijven $C(n)$ voor de maximale lengte van het kortste synchroniserende woord voor een synchroniseerbare automaat met n toestanden.

Stelling van Frankl en Pin (1983)

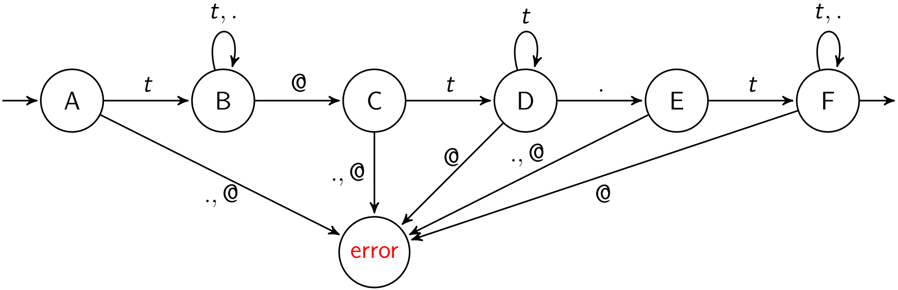
Voor iedere n geldt: $C(n) \leq \dfrac{n^3 - n}{6}$.

- Het bewijs is niet makkelijk. Op de handout: $C(n) \leq \frac{n^3-2n^2+n}{2}$.
- Er zijn betere resultaten voor *speciale soorten automaten*.

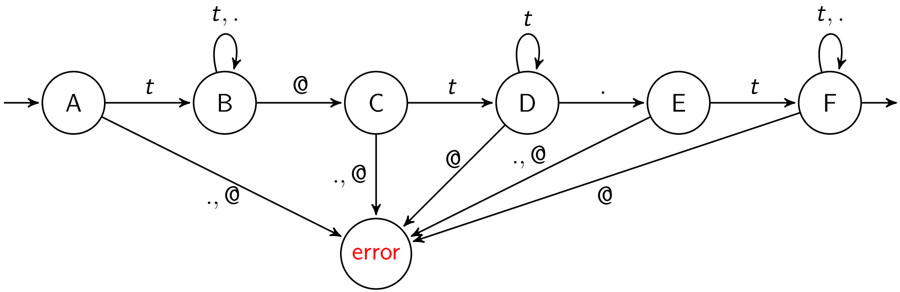
Grenzen voor $C(n)$



De automaat voor e-mailadressen

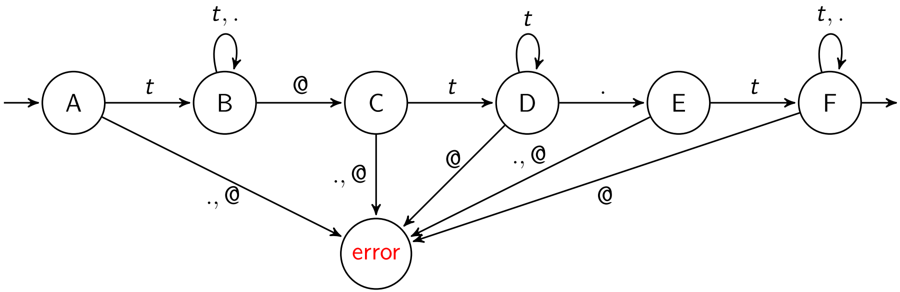


De automaat voor e-mailadressen



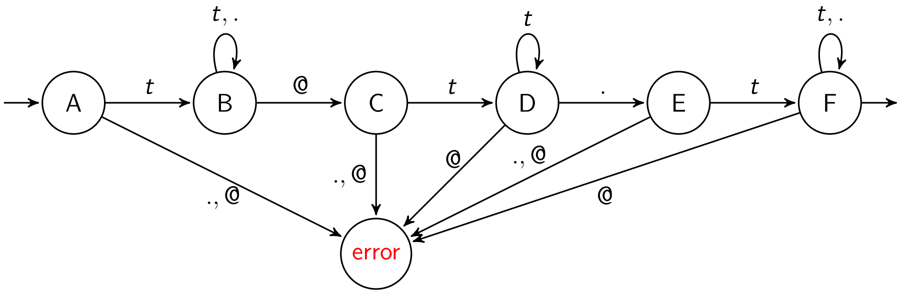
Als we het woord 'S.J.vanGool@uva.nl' invoeren in toestand A, eindigen we in toestand F.

De automaat voor e-mailadressen



Als we het woord ‘S.J.vanGool@uva.nl’ invoeren in toestand A, eindigen we in toestand F. Als we het woord ‘sam@van@gool’ invoeren in toestand A, eindigen we in toestand error.

De automaat voor e-mailadressen



Als we het woord 'S.J.vanGool@uva.nl' invoeren in toestand A, eindigen we in toestand F. Als we het woord 'sam@van@gool' invoeren in toestand A, eindigen we in toestand error.
De automaat *accepteert* het woord 'S.J.vanGool@uva.nl' en *wijst* het woord 'sam@van@gool' *af*.

Woorden accepteren en afwijzen

- De automaat voor e-mailadressen had als doel: geldige e-mailadressen accepteren, ongeldige e-mailadressen afwijzen.

Woorden accepteren en afwijzen

- De automaat voor e-mailadressen had als doel: geldige e-mailadressen accepteren, ongeldige e-mailadressen afwijzen.
- We zeggen: de automaat *herkent* de verzameling geldige e-mailadressen.

Woorden accepteren en afwijzen

- De automaat voor e-mailadressen had als doel: geldige e-mailadressen accepteren, ongeldige e-mailadressen afwijzen.
- We zeggen: de automaat *herkent* de verzameling geldige e-mailadressen.

Reguliere talen

- Een verzameling woorden heet een *reguliere taal* als zij herkend kan worden door een automaat.

Reguliere talen

- Een verzameling woorden heet een *reguliere taal* als zij herkend kan worden door een automaat.

Voorbeeld

- De verzameling *geldige e-mailadressen* is een reguliere taal.

Reguliere talen

- Een verzameling woorden heet een *reguliere taal* als zij herkend kan worden door een automaat.

Voorbeeld

- De verzameling *geldige e-mailadressen* is een reguliere taal.
- De verzameling *ongeldige e-mailadressen* is ook een reguliere taal. (Waarom?)

Reguliere expressies

- Een compactere omschrijving van de reguliere taal ‘geldige e-mailadressen’ is de volgende *reguliere expressie*:

$$t[\neg \texttt{@}]^*\texttt{@}[t]^+.[t,.]^+$$

Reguliere expressies

- Een compactere omschrijving van de reguliere taal ‘geldige e-mailadressen’ is de volgende *reguliere expressie*:

$$t[\neg\texttt{@}]^*\texttt{@}[t]^+.[t,.]^+$$

  - het symbool $\neg$ betekent ‘niet’;

Reguliere expressies

- Een compactere omschrijving van de reguliere taal 'geldige e-mailadressen' is de volgende *reguliere expressie*:

$$t[\neg \texttt{@}]^*\texttt{@}[t]^+.[t,.]^+$$

  - het symbool $\neg$ betekent 'niet';
  - het sterretje * betekent: een willekeurig aantal herhalingen (mogelijk nul) van wat binnen de haken staat;

Reguliere expressies

- Een compactere omschrijving van de reguliere taal ‘geldige e-mailadressen’ is de volgende *reguliere expressie*:

$$t[\neg @]^{*}@[t]^{+}.[t,.]^{+}$$

  - het symbool $\neg$ betekent ‘niet’;
  - het sterretje * betekent: een willekeurig aantal herhalingen (mogelijk nul) van wat binnen de haken staat;
  - de plus $^{+}$ betekent: een positief (minstens één) aantal herhalingen;

Reguliere expressies

- Een compactere omschrijving van de reguliere taal ‘geldige e-mailadressen’ is de volgende *reguliere expressie*:

$$t[\neg @]^{*}@[t]^{+}.[t,.]^{+}$$

  - het symbool $\neg$ betekent ‘niet’;
  - het sterretje * betekent: een willekeurig aantal herhalingen (mogelijk nul) van wat binnen de haken staat;
  - de plus $^{+}$ betekent: een positief (minstens één) aantal herhalingen;
  - en de komma , betekent: de ene óf de andere expressie.

Reguliere expressies

- Een compactere omschrijving van de reguliere taal ‘geldige e-mailadressen’ is de volgende *reguliere expressie*:

$$t[\neg\texttt{@}]^{*}\texttt{@}[t]^{+}.[t,.]^{+}$$

  - het symbool $\neg$ betekent ‘niet’;
  - het sterretje * betekent: een willekeurig aantal herhalingen (mogelijk nul) van wat binnen de haken staat;
  - de plus $^{+}$ betekent: een positief (minstens één) aantal herhalingen;
  - en de komma , betekent: de ene óf de andere expressie.
- Reguliere expressies kunnen worden gevormd door letters, negaties, sterretjes, plussen en komma’s naast elkaar te zetten.

Reguliere expressies

- Een compactere omschrijving van de reguliere taal ‘geldige e-mailadressen’ is de volgende *reguliere expressie*:

$$t[\neg @]^* @[t]^+ .[t, .]^+$$

  - het symbool $\neg$ betekent ‘niet’;
  - het sterretje * betekent: een willekeurig aantal herhalingen (mogelijk nul) van wat binnen de haken staat;
  - de plus $^+$ betekent: een positief (minstens één) aantal herhalingen;
  - en de komma , betekent: de ene óf de andere expressie.
- Reguliere expressies kunnen worden gevormd door letters, negaties, sterretjes, plussen en komma’s naast elkaar te zetten.

Stelling

Elke reguliere taal kan worden omschreven door een reguliere expressie, en elke reguliere expressie geeft een reguliere taal.

Een niet-reguliere taal: twee keer hetzelfde wachtwoord

- Bij het aanmaken van een nieuw wachtwoord moet je *twee keer hetzelfde* wachtwoord invullen, met daartussen één keer de toets 'enter' (`ENT`).

Een niet-reguliere taal: twee keer hetzelfde wachtwoord

- Bij het aanmaken van een nieuw wachtwoord moet je *twee keer hetzelfde* wachtwoord invullen, met daartussen één keer de toets 'enter' (`ENT`).
- We willen dus een automatisch proces dat alleen invoer van de vorm $w\texttt{ENT}w$ accepteert, waar w een geldig wachtwoord is.

Een niet-reguliere taal: twee keer hetzelfde wachtwoord

- Bij het aanmaken van een nieuw wachtwoord moet je *twee keer hetzelfde* wachtwoord invullen, met daartussen één keer de toets 'enter' (`ENT`).
- We willen dus een automatisch proces dat alleen invoer van de vorm $w\texttt{ENT}w$ accepteert, waar w een geldig wachtwoord is.
- De taal van geldige wachtwoorden is een reguliere taal.

Een niet-reguliere taal: twee keer hetzelfde wachtwoord

- Bij het aanmaken van een nieuw wachtwoord moet je *twee keer hetzelfde* wachtwoord invullen, met daartussen één keer de toets 'enter' (`ENT`).
- We willen dus een automatisch proces dat alleen invoer van de vorm $w\texttt{ENT}w$ accepteert, waar w een geldig wachtwoord is.
- De taal van geldige wachtwoorden is een reguliere taal.
- De taal bestaande uit woorden van de vorm $w\texttt{ENT}w$ is géén reguliere taal.

Een niet-reguliere taal: twee keer hetzelfde wachtwoord

- Bij het aanmaken van een nieuw wachtwoord moet je *twee keer hetzelfde* wachtwoord invullen, met daartussen één keer de toets 'enter' (`ENT`).
- We willen dus een automatisch proces dat alleen invoer van de vorm $w\texttt{ENT}w$ accepteert, waar w een geldig wachtwoord is.
- De taal van geldige wachtwoorden is een reguliere taal.
- De taal bestaande uit woorden van de vorm $w\texttt{ENT}w$ is géén reguliere taal. Hoe kun je dat bewijzen?

Een niet-reguliere taal: twee keer hetzelfde wachtwoord

- Bij het aanmaken van een nieuw wachtwoord moet je *twee keer hetzelfde* wachtwoord invullen, met daartussen één keer de toets 'enter' (`ENT`).
- We willen dus een automatisch proces dat alleen invoer van de vorm $w\texttt{ENT}w$ accepteert, waar w een geldig wachtwoord is.
- De taal van geldige wachtwoorden is een reguliere taal.
- De taal bestaande uit woorden van de vorm $w\texttt{ENT}w$ is géén reguliere taal. Hoe kun je dat bewijzen?
- Reguliere talen hebben een *lage complexiteit*.

Een niet-reguliere taal: twee keer hetzelfde wachtwoord

- Bij het aanmaken van een nieuw wachtwoord moet je *twee keer hetzelfde* wachtwoord invullen, met daartussen één keer de toets 'enter' (`ENT`).
- We willen dus een automatisch proces dat alleen invoer van de vorm $w\texttt{ENT}w$ accepteert, waar w een geldig wachtwoord is.
- De taal van geldige wachtwoorden is een reguliere taal.
- De taal bestaande uit woorden van de vorm $w\texttt{ENT}w$ is géén reguliere taal. Hoe kun je dat bewijzen?
- Reguliere talen hebben een *lage complexiteit*.
- "Automaten zijn computers met weinig geheugenruimte".

Ster-vrije reguliere expressies

- In een *ster-vrije* reguliere expressie mag je geen sterretjes en geen plussen gebruiken.

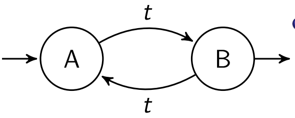
Ster-vrije reguliere expressies

- In een *ster-vrije* reguliere expressie mag je geen sterretjes en geen plussen gebruiken.
- Dus alleen nog letters, negatie ($\neg$), de komma (,), en het symbool $\top$, dat staat voor 'alle woorden'.

Ster-vrije reguliere expressies

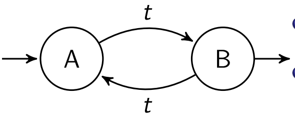
- In een *ster-vrije* reguliere expressie mag je geen sterretjes en geen plussen gebruiken.
- Dus alleen nog letters, negatie ($\neg$), de komma (,), en het symbool $\top$, dat staat voor 'alle woorden'.
- Sommige reguliere expressies kunnen worden omgeschreven naar ster-vrije reguliere expressies:
  - De expressie $[\neg\texttt{@}]^*$ beschrijft dezelfde taal als $\neg[\top\texttt{@}\top]$.
  - De expressie $[t]^+$ beschrijft dezelfde taal als $\neg[\top[.,\texttt{@}]\top, \neg[\top[.,\texttt{@},t]\top]]$.

Wanneer bestaat er een ster-vrije expressie?



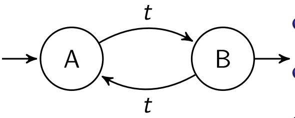
- Wat is de taal die deze automaat herkent?

Wanneer bestaat er een ster-vrije expressie?



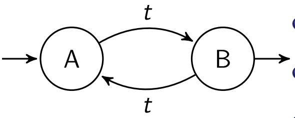
- Wat is de taal die deze automaat herkent?
- Wat is een reguliere expressie voor deze taal?

Wanneer bestaat er een ster-vrije expressie?



- Wat is de taal die deze automaat herkent?
- Wat is een reguliere expressie voor deze taal?
- Bestaat er een *ster-vrije* expressie voor de taal?

Wanneer bestaat er een ster-vrije expressie?

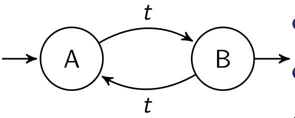


- Wat is de taal die deze automaat herkent?
- Wat is een reguliere expressie voor deze taal?
- Bestaat er een *ster-vrije* expressie voor de taal?

Stelling (Schützenberger 1965)

Een reguliere taal heeft een ster-vrije expressie dan, en slechts dan, als er een **aperiodieke** *automaat voor de taal bestaat.*

Wanneer bestaat er een ster-vrije expressie?



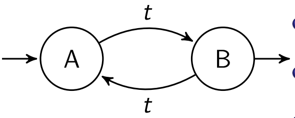
- Wat is de taal die deze automaat herkent?
- Wat is een reguliere expressie voor deze taal?
- Bestaat er een *ster-vrije* expressie voor de taal?

Stelling (Schützenberger 1965)

Een reguliere taal heeft een ster-vrije expressie dan, en slechts dan, als er een **aperiodieke** *automaat voor de taal bestaat.*

Een automaat heet **aperiodiek** als er een n bestaat zo dat voor ieder woord w geldt dat w^n en w^{n+1} precies hetzelfde effect hebben.

Wanneer bestaat er een ster-vrije expressie?



- Wat is de taal die deze automaat herkent?
- Wat is een reguliere expressie voor deze taal?
- Bestaat er een *ster-vrije* expressie voor de taal?

Stelling (Schützenberger 1965)

Een reguliere taal heeft een ster-vrije expressie dan, en slechts dan, als er een **aperiodieke** *automaat voor de taal bestaat.*

Een automaat heet **aperiodiek** als er een n bestaat zo dat voor ieder woord w geldt dat w^n en w^{n+1} precies hetzelfde effect hebben.
"*Alle ondergroepen van de halfgroep voortgebracht door de transitiefuncties zijn triviaal.*"

Černý's vermoeden voor aperiodieke automaten

Stelling (Trahtman 2007)

Voor iedere **aperiodieke** *synchroniseerbare automaat met n toestanden bestaat er een synchroniserend woord van lengte hoogstens $(n-1)^2$.*

Conclusie

- Automaten zijn de eenvoudigste soort rekenmachines.

Conclusie

- Automaten zijn de eenvoudigste soort rekenmachines.
- Hoe eenvoudig ook, ze bieden een rijke bron aan wiskundige problemen en theorie.

Conclusion

- Automaten zijn de eenvoudigste soort rekenmachines.
- Hoe eenvoudig ook, ze bieden een rijke bron aan wiskundige problemen en theorie.
- Automatentheorie bevindt zich op het grensvlak van de *informatica*, (lineaire) *algebra*, *combinatoriek*, en *logica*, met soms ook wat *topologie*.

Conclusie

- Automaten zijn de eenvoudigste soort rekenmachines.
- Hoe eenvoudig ook, ze bieden een rijke bron aan wiskundige problemen en theorie.
- Automatentheorie bevindt zich op het grensvlak van de *informatica*, (lineaire) *algebra*, *combinatoriek*, en *logica*, met soms ook wat *topologie*.
- De slides, met achterin hints & bronnen, staan straks online: http://www.samvangool.net/ldw2018.pdf

Alle wegen leiden naar automaten

Sam van Gool

Institute for Logic, Language and Computation
Universiteit van Amsterdam

Leve De Wiskunde! Symposium
6 april 2018

Meer lezen

- Een goed leesbare inleiding tot automatentheorie (en meer):

  P. Linz, An Introduction to Formal Languages and Automata, Jones & Bartlett Learning (2016).

- Over het vermoeden van Černý:

  M. Volkov, Synchronizing Automata and the Černý Conjecture. In: LATA 2008, LNCS 5196, 11–27 (2008).

- Over de stelling van Schützenberger:

  T. Colcombet. Greens Relations and their Use in Automata Theory. In: LATA 2011, LNCS 6638, 1–21 (2011).

- Twee grote standaardwerken over halfgroepentheorie:

  J. Rhodes and B. Steinberg, The q-theory of Finite Semigroups, Springer (2009).

  B. Steinberg, Representation Theory of Finite Monoids, Springer (2016).

Bronvermelding

- De vlaggen-truuk is gebaseerd op: http://www.realmagic.net/dp/1-3.htm
- Het grote schema voor e-mailadressen komt van: http://emailregex.com/
- Het functie-plot is gemaakt met: http://www.fooplot.com

Instructies voor de vlaggen-truuk

- Begin in een vlag waar de kleur blauw in voorkomt.
- Beweeg naar links of rechts naar een vlag waar groen in voorkomt.
- Beweeg omhoog of omlaag naar een vlag waar geel in voorkomt.
- Beweeg naar links of omlaag naar een vlag waar groen in voorkomt.

Hints voor de opgaven uit de lezing

- Voor de doolhofpuzzel: bekijk opnieuw wat *RGGG* doet. Wat gebeurt er als je dit herhaalt? Nu heb je 8 letters, wat is de negende?
- Voor de doolhoven met meer kamers: in het geval van 4 kamers werkte *RGGG*. Wat kun je doen om iets soortgelijks te bereiken met 5 kamers? Je mag 1 letter toevoegen.
- Om de e-mailautomaat aan te passen zoals gevraagd in de handout: je hebt maar 1 extra toestand nodig. Denk: als je voor het apenstaartje nog een punt leest, dan heb je *misschien* een error, namelijk wanneer? En wanneer zul je toch geen error hebben?

Hints voor de verdiepingsopgaven

- Om een pomp-getal voor een reguliere taal te vinden, bekijk je een automaat die de taal herkent. Als de automaat n toestanden heeft, dan werkt $m = n + 1$ altijd als pomp-getal – waarom?
- Om te laten zien dat de taal van herhaalde wachtwoorden niet-regulier is, wil je voor ieder mogelijk pomp-getal $m \geq 1$ een nog *langer* herhaald wachtwoord kiezen, waarvan de gepompte versies dan niet meer 'correct herhaald' worden.
- Merk op dat de grens voor synchroniserende woorden gelijk is aan $(n-1)\binom{n}{2}$.
- Voor het antwoord op de moeilijke stap (d) in de laatste verdiepingsopgave, zie de volgende slide.

Bewijs van grens voor synchroniseren van twee toestanden

- In dit bewijs korten we $\delta_w(q)$ voor het gemak af als 'qw'.
- Kies een woord $w = a_1 \dots a_m$ van minimale lengte zo dat $qw = rw$.
- Voor $0 \leq i < m$ schrijven we w_i voor de prefix van w van lengte i, dus w_0 is het lege woord en $w_i = a_1 \dots a_i$.
- We zullen laten zien dat er *geen* $0 \leq i < j < m$ bestaan zodat $\{qw_i, rw_i\} = \{qw_j, rw_j\}$.
- We redeneren met tegenspraak. Stel dat er wél zulke i en j bestaan.
- We kunnen het woord w dan schrijven als $w = w_i \alpha v_j$, waar $\alpha = a_{i+1} \dots a_j$ en $v_j = a_{j+1} \dots a_m$. Merk op dat het woord α niet leeg is, en dat $w = w_j v_j$.
- Er zijn 2 gevallen: (i) $qw_i = qw_j$, $rw_i = rw_j$, of (ii) $qw_i = rw_j$, $rw_i = qw_j$.
- In beide gevallen geldt nu dat $qw_i v_j = rw_i v_j$:
  - In geval (i) hebben we $qw_i v_j = qw_j v_j = qw = rw = rw_j v_j = rw_i v_j$.
  - In geval (ii) hebben we $qw_i v_j = rw_j v_j = rw = qw = qw_j v_j = rw_i v_j$.
- We hebben laten zien dat het woord $w' = w_i v_j$ óók de eigenschap $qw' = rw'$ heeft, hetgeen de aanname dat w van minimale lengte was tegenspreekt.
- Dus $\{q, r\}, \{qw_1, rw_1\}, \dots, \{qw_{n-1}, rw_{m-1}\}$ zijn m verschillende paren toestanden. Er zijn slechts $\binom{n}{2}$ zulke paren, dus $m \leq \binom{n}{2}$.