



Radboud Universiteit Nijmegen

BACHELORSCHRIPTIE WISKUNDE
SYNCHRONISERENDE AUTOMATEN

Begeleider:

Dhr. Dr. ir. H. Don

Tweede lezer:

Dhr. Dr. W. Bosma

Student:

A.R. Jansen s4583124

Juli 2019

Inhoudsopgave

1	Inleiding	2
2	Automaten	3
3	Černý's vermoeden	6
3.1	Boven- en ondergrenzen	6
3.2	Černý automaten	10
4	Random input	13
4.1	Markovketens	13
4.2	Berekening verwachte lengte van een synchroniserend woord	14
4.3	Synchronisatie Černý automaat	16
4.4	Synchronisatie van andere automaten	22
5	Resultaten	25
5.1	Černý automaat	25
5.2	Slechte automaat	25
5.3	Andere automaten	25
5.4	Conclusie	25

1 Inleiding

Het onderwerp van mijn scriptie is, zoals te lezen valt in de titel, synchroniserende automaten. De volgende vakgebieden staan hierin centraal: combinatoriek, discrete wiskunde, kansrekening en theoretische computer science.

Ik heb gekozen voor dit onderwerp, omdat ik in mijn derde jaar het vak talen en automaten bij de studie Informatica gevolgd heb. Het vak sprak mij heel erg aan. Het lijkt heel erg op grafentheorie en je kan er heel goed kansrekeningen op toepassen, zoals ik ook in mijn scriptie heb gedaan.

Het vak talen en automaten is geen vereiste om mijn scriptie te begrijpen. In de tweede sectie worden alle benodigde definities uitgelegd.

In de derde sectie staat het vermoeden van Černý centraal. Dit vermoeden bedacht door Jan Černý, geeft ons inzicht in de maximale lengte van het kortste synchroniserende woord voor een automaat op n toestanden. Zoals het woord vermoeden al zegt is er tot op heden nog geen bewijs gevonden voor dit vermoeden. Wel is er een speciale klasse van automaten bekend waarvoor de grens van het vermoeden gehaald wordt: de Černý automaten. Deze automaat wordt uitvoerig besproken in sectie 3.2.

In sectie 4 gaan we ons meer richten op de stochastiek van automaten. Waar we in de eerste- en tweede-sectie keken naar deterministische input richten we ons in de vierde sectie op de random input. Dit verandert de kijk op synchronisatie. We zijn in deze sectie vooral geïnteresseerd in de verwachte lengte van een woord tot we een synchroniserende automaat hebben gesynchroniseerd.

In de laatste sectie staan de gevonden resultaten centraal. Wat zegt de kortste lengte van een synchroniserend woord voor een automaat over de verwachte tijd die nodig is om die automaat te synchroniseren met random input? Zijn de Černý automaten ook met random input het moeilijkst te synchroniseren?

2 Automaten

Automaattheorie is de studie van automaten en is een vakgebied dat ligt in de theoretische computer Science en de discrete Wiskunde.

In deze sectie geven we de benodigde voorkennis over automaten. Ook leggen we in deze sectie de notatie voor automaten vast.

Nog voordat we de definitie van een automaat geven is het van belang dat we de volgende definities hebben [1]:

Definitie 2.1. Een alfabet Σ is een eindige verzameling van symbolen bijvoorbeeld letters, tekens of cijfers.

Voorbeeld 2.2. Een paar voorbeelden van een alfabet zijn:

- $\{a, b, c, \dots, z\}$
- $\{1, 2, 3, 4\}$
- $\{\alpha, \beta, \dots, \zeta\}$

Definitie 2.3. Een woord w is een eindige rij van symbolen uit een alfabet Σ . Σ^* is de verzameling van alle woorden die kunnen worden gemaakt met het alfabet Σ . We definiëren λ als het lege woord.

De automaten waar wij het over gaan hebben zijn deterministische eindige automaten.

Definitie 2.4. Een deterministische eindige automaat (DFA) $\mathcal{A}$ is een 3-tupel (Q, Σ, δ) met:

Q : (eindige) Verzameling van toestanden.

Σ : Het alfabet van de automaat.

$\delta: \delta: Q \times \Sigma \longrightarrow Q$ is de transitie functie, waarbij de uitkomst weer een toestand is.

$\delta(q, l) = q'$ dat wil zeggen dat je met behulp van letter $l \in \Sigma$ van toestand q naar toestand q' komt.

Wij zijn geïnteresseerd in woorden. Het is dus van belang dat wij de transitie functie uitbreiden naar een functie voor woorden. Als we de transitie functie uitbreiden krijgen we de volgende functie.

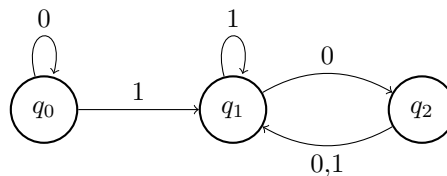
Definitie 2.5. Neem $w \in \Sigma^*$ met $w \neq \lambda$ waarvoor geldt dat $w = vl$ met $v \in \Sigma^*, l \in \Sigma$. Neem $x \in Q$

We definiëren de functie als volgt:

$$\delta: Q \times \Sigma^* \longrightarrow Q$$

$$\delta(x, w) = \delta(\delta(x, v), l)$$

Er geldt dat $\delta(x, \lambda) = x$



Figuur 1: Deterministische eindige automaat

Voorbeeld 2.6. In figuur 1 zien we een deterministische automaat, hiervoor geldt :

$$Q = \{q_0, q_1, q_2\}$$

$$\Sigma = \{0, 1\}$$

$$\delta(q_0, 0) = q_0$$

$$\delta(q_0, 1) = q_1$$

$$\delta(q_1, 1) = q_1$$

$$\delta(q_1, 0) = q_2$$

$$\delta(q_2, 0) = q_1$$

$$\delta(q_2, 1) = q_1$$

$$\delta(q_0, 0110) = \delta(\delta(q_0, 011), 0) = \delta(\delta(\delta(q_0, 01), 1), 0) = \delta(\delta(\delta(\delta(q_0, 0), 1), 1), 0) = q_2$$

Een definitie die ook van belang is, is die van de powerautomaat. Deze automaat wordt in het vervolg nog veel gebruikt.

Definitie 2.7. Een powerautomaat $\mathcal{P}_A$ van een DFA $\mathcal{A} = (Q, \Sigma, \delta)$ is een 3-tupel $(\bar{Q}, \bar{\Sigma}, \bar{\delta})$ met:

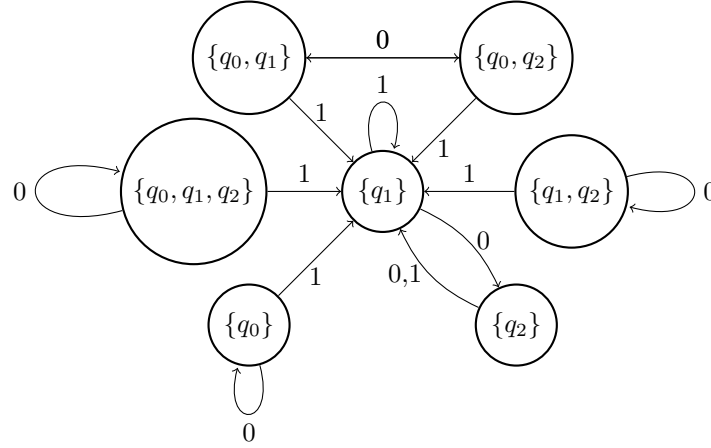
$$\bar{Q} = \mathcal{P}(Q) \setminus \emptyset$$

$$\bar{\Sigma} = \Sigma$$

$$\bar{\delta}(H, w) = \bigcup_{q \in H} \delta(q, w) \text{ met } H \subseteq Q, H \neq \emptyset.$$

Opmerking: Een powerautomaat is zelf ook een DFA.

Voorbeeld 2.8. De powerautomaat van de automaat gegeven in figuur 1, is weergegeven in figuur 2.



Figuur 2: Powerautomaat

Wij zijn geïnteresseerd in synchroniserende automaten, maar wat is nu een synchroniserende automaat?

Definitie 2.9. Een automaat $\mathcal{A} = (Q, \Sigma, \delta)$ heet synchroniserend als $\exists w \in \Sigma^*$ en een $q \in Q$ bestaat zodanig dat $\forall q' \in Q, \delta(q', w) = q$. Het woord w wordt dan synchroniserend genoemd.

Opmerking: Niet elke automaat is synchroniserend.

Opmerking: Als een automaat synchroniserend is, dan heeft hij meerdere synchroniserende woorden. Een manier om synchroniserende woorden te vinden is met behulp van de powerautomaat

Definitie 2.10. Een woord $w \in \Sigma^*$ heet synchroniserend dan en slechts dan als w een pad geeft in de powerautomaat $\mathcal{P}(\mathcal{A})$ van Q naar een een-punt-verzameling.

Op de bovenstaande manier een synchroniserend woord maken is conceptueel erg makkelijk, maar is erg inefficiënt omdat de powerautomaat $\mathcal{P}(\mathcal{A})$ exponentieel groter is dan $\mathcal{A}$.

Voorbeeld 2.11. Zie figuur 3. Deze automaat op vier toestanden is synchroniserend. Een synchroniserend woord dat bij deze automaat hoort is de volgende ab^3ab^3a , want $\delta(i, ab^3ab^3a) = 1 \forall i \in Q$. Een ander synchroniserend woord is bijvoorbeeld ab^2abab^3a .

Voorbeeld 2.12. Zie figuur 4, voor de powerautomaat, die hoort bij de automaat uit figuur 3. Dan zien we dat ab^33ab^3a inderdaad een pad is van $Q = \{0, 1, 2, 3\}$ naar een een-punt-verzameling.

We zien dat ab^2abab^3a ook een pad is van $Q = \{0, 1, 2, 3\}$ naar een een-punt-verzameling, maar dit woord is erg greedy, het wil zo snel mogelijk naar een toestand met een kleinere verzameling. Dit heeft helaas niet tot gevolg dat het het kortste synchroniserende woord is geworden.

3 Černý's vermoeden

In deze sectie staat het vermoeden van Černý centraal. Voordat we kunnen beschrijven wat het vermoeden is, hebben we een definitie nodig.

Definitie 3.1. *Zij $\mathcal{A}$ een synchroniserende automaat met n toestanden, en zij w het kortste synchroniserende woord laat dan $l(\mathcal{A}) = |w|$. Definieer $C(n)$ als $C(n) = \max\{l(\mathcal{A}) \mid \mathcal{A} \text{ synchroniseert en heeft } n \text{ toestanden}\}$*

In 1964 kwam Černý met een vermoeden over de waarde van $C(n)$. Černý's vermoeden zegt het volgende[2]:

$$C(n) = (n - 1)^2 \quad (1)$$

Anders gezegd: elke synchroniserende automaat met n toestanden heeft een synchroniserend woord met lengte $\leq (n - 1)^2$

3.1 Boven- en ondergrenzen

Hoewel het vermoeden van Černý nog niet is bewezen, zijn er wel onder- en bovengrenzen gevonden die iets zeggen over de waarde van $C(n)$.

Černý kwam in 1964 zelf met de volgende ondergrens:

$$C(n) \geq (n - 1)^2 \quad (2)$$

Deze ondergrens is gebaseerd op synchroniserende woorden in een speciale automaat: Černý's automaat $\mathcal{C}_n$. In figuur 3 hebben we al een voorbeeld gezien van deze speciale klasse van automaten. We gaan later dieper in op deze klasse van automaten.

Černý kwam in 1964 zelf ook met een bovengrens voor $C(n)$ namelijk:

$$C(n) \leq 2^n - n - 1 \quad (3)$$

Lemma 3.2. *Zij $n \in \mathbb{N}$ dan $C(n) \leq 2^n - n - 1$.*

Bewijs. Laat $\mathcal{A}$ een deterministische eindige automaat op n toestanden, $n \in \mathbb{N}$, dan bekijken we de power automaat $\mathcal{P}(\mathcal{A})$. Deze heeft $2^n - 1$ toestanden, de lege verzameling is weggelaten. We zijn op zoek naar de maximale lengte van het kortste woord, we zijn dus eigenlijk opzoek naar de maximale lengte van het kortste pad van $\{1, \dots, n\}$ naar een een-punt-verzameling. Hiervoor moeten we door alle toestanden, behalve de een-punt-verzamelen, ten hoogste één keer gaan. Dus de maximale lengte van zo een pad is: $2^n - n - 1$, want er zijn $2^n - 1$ toestanden in de powerautomaat, en er zijn n toestanden die horen bij een een-punt-verzameling. Dus de maximale lengte van het kortste synchroniserende woord is: $2^n - n - 1$ $\square$

In 1966 verbeterde Starke deze bovengrens. De bovengrens die hij had gevonden was:

$$C(n) \leq 1 + \frac{n(n-1)(n-2)}{2}. \quad (4)$$

Lemma 3.3. *Zij $n \in \mathbb{N}$ dan $C(n) \leq 1 + \frac{n(n-1)(n-2)}{2}$.*

Bewijs. Laat $\mathcal{A}$ een deterministische eindige automaat op n toestanden, $n \in \mathbb{N}$. Als we gaan kijken naar 2 toestanden $p, q \in Q$ en we hebben een synchroniserend woord $w \in \Sigma^*$ voor deze 2 toestanden, dat wil zeggen dat $\delta(p, w) = \delta(q, w)$. Dan kunnen we w zo kiezen zodat $|w| \leq \binom{n}{2}$, want als we de kortste lengte willen van w dan moet gelden dat je maximaal één keer door een toestand die hoort bij een paar gaat in de powerautomaat. $\mathcal{P}(\mathcal{A})$. Want als we beginnen in de toestand die hoort bij het paar $\{p, q\}$ dan kunnen we nooit meer met een pad naar een toestand van een verzameling van meer dan 2 elementen, dus we gaan een keer door alle toestanden die horen bij een paar en dan naar een een-punt-verzameling.

We kunnen op $|w| \leq \binom{n}{2}$ manieren maximaal één keer door elke toestand, behorend bij een paar, in de powerautomaat $\mathcal{P}(\mathcal{A})$.

We nemen $S \subseteq Q$ en stel dat $p, q \in S$ en er is een synchroniserend woord $w \in \Sigma^*$ voor p, q . Definieer dan S_w als $\bar{\delta}(S, w)$, dat wil zeggen dat S_w de verzameling toestanden is die we bereiken als we het woord w toepassen op de verzameling S . Er geldt dan dat $|S_w| < |S|$, want p, q worden naar dezelfde toestand gestuurd in S_w en in het slechtste geval worden alle andere toestanden naar iets anders gestuurd dan geldt in het slechtste geval dat $|S_w| = |S| - 1$.

We gaan nu itereren over S .

Neem eerst $S = Q$ dan hebben we sowieso een woord van lengte 1 nodig om naar een lagere toestand te gaan in de powerautomaat. Kijk nu naar $S \subset Q$ dan geldt er dat $|w| \leq \binom{n}{2}$ en dit geldt voor alle $n - 2$ keren dat we dit itereren. voor $|S| = 2, \dots, n - 1$, tot je bij $S = 1$ bent.

Dit bij elkaar genomen geldt dat: $|w| \leq 1 + \binom{n}{2}(n - 2) = 1 + \frac{n(n-1)(n-2)}{2}$. Dus $C(n) \leq 1 + \frac{n(n-1)(n-2)}{2}$ $\square$

Het duurde vervolgens een paar jaar tot er een nieuwe verbeterde bovengrens werd gevonden. Deze werd gevonden door Frankl die gebruik heeft gemaakt van de inzichten van Pin. De bovengrens die hij had gevonden was:

$$C(n) \leq \frac{n^3 - n}{6}. \quad (5)$$

Stelling 3.4. *Zij $n \in \mathbb{N}$ dan $C(n) \leq \frac{n^3 - n}{6}$.*

Pin heeft dit lemma gereduceerd tot een ander probleem. Dat probleem is geformuleerd als stelling door Pin en bewezen door Frankl. [3] [2]

Stelling 3.5. *Stel $\mathcal{A} = \{A_1, \dots, A_m\}$ is een familie van r -verzamelingen, $\mathcal{B} = \{B_1, \dots, B_m\}$ is een familie van s -verzamelingen zodanig dat*

- $A_i \cap B_i = \emptyset$ for $i = 1, \dots, m$
- $A_i \cap B_j \neq \emptyset$ for $i \leq i < j \leq m$

dan

$$m \leq \binom{r+s}{s}.$$

Om deze stelling te bewijzen hebben we het volgende lemma nodig.

Lemma 3.6. *Stel we hebben s ballen die we moeten verdelen over $r + 1$ bakken, met $s, r \in \mathbb{N}$. Dan kan dit op $\binom{r+s}{s}$ manieren*

Bewijs. Van lemma 3.6 We hebben hier te maken met herhalingscombinaties. Het maakt namelijk niet uit hoeveel ballen er in een bak komen en op welke volgorde de bakken worden gevuld.

We kunnen de s ballen zien als rij van bijvoorbeeld stipjes Om deze te verdelen in $r + 1$ bakken, moeten we deze rij van s stipjes opsplitsen in $r + 1$ groepen, want elke groep kan dan worden geplaatst in één van de bakken.

Hoe vinden we nu die groepen? We kunnen hiervoor $(r + 1) - 1$ scheidingen maken tussen de rij stipjes. Stel zo een scheiding voor als bijvoorbeeld een streep. Als we dan de stipjes en streepjes in een rij naast elkaar hebben staan staan er $(r + 1) - 1 + s$ tekens, waarvan er s stipjes moeten zijn en $(r + 1) - 1$ een streep. Op hoeveel manieren kunnen we van die $(r + 1) - 1 + s$ tekens er s kiezen die een stip moeten zijn? Dit kan op $\binom{(r+1)-1+s}{s} = \binom{r+s}{s}$ manieren. Als de stipjes vaststaan, dan staan ook de streepjes vast. Dus we kunnen op $\binom{r+s}{s}$ de s ballen verdelen over $r + 1$ bakken. $\square$

Bewijs. Bewijs van stelling 3.5

Neem $X = \bigcup_{i=1}^m (A_i \cup B_i)$, dan is X een eindige verzameling. Immers alle A_i 's en B_i 's zijn eindig. Hieruit mogen we aannemen dat alle punten uit X in $\mathbb{R}^{r+1}$ liggen. Nog belangrijker de punten van X en de oorsprong liggen in generale positie. Dat wil zeggen dat het opspansel van elke $r + 1$ punten een hyper-vlak van dimensie r is en er geen andere punten tussenin bevat.

In het bijzonder voor $i = 1, \dots, m$ spannen de punten van A_i en de oorsprong een deelruimte van dimensie r op. We noemen deze deelruimte V_i . Er geldt voor deze V_i dat $V_i \cap (X \setminus A_i) = \emptyset$.

Voor het gemak maken we geen onderscheid tussen een punt in $\mathbb{R}^{r+1}$ en zijn vector (dat wil zeggen de vector tussen de oorsprong en het punt in $\mathbb{R}^{r+1}$).

Definieer u_i als de normaal eenheid-vector van V_i . Dat wil zeggen dat u_i orthogonaal is aan V_i en zijn punten aan de positieve zijde van V_i . Er geldt dus dat $a \in V_i \Leftrightarrow (a, u_i) = 0$.

Laat nu f_a de lineaire functionaal gedefinieerd door $f_a(v) = (a, v)$, dan is de ruimte van lineaire functionalen isomorf met $\mathbb{R}^{r+1}$. Het isomorfisme wordt gegeven met de volgende functie.

Definieer F als de ruimte van lineaire functionalen.

$$\phi : \mathbb{R}^{r+1} \longrightarrow F : a \mapsto f_a \quad (6)$$

De inverse wordt de volgende functie:

$$\phi^{-1} : F \longrightarrow \mathbb{R}^{r+1} : f_a \mapsto a \quad (7)$$

Hieruit volgt dat:

$\phi(\phi^{-1}(f_a)) = \phi(a) = f_a$ dus wordt de identiteit functie op F

$\phi^{-1}(\phi(a)) = \phi^{-1}(f_a) = a$ dus wordt de identiteit functie op $\mathbb{R}^{r+1}$

Definieer nu $W_1, \dots, W_s$ als de s disjuncte kopieën van F .

Het tensorproduct van $W_1, \dots, W_s$ is de vectorruimte van multi-lineaire functies. Dat wil zeggen dat de functie lineair is in elk van zijn variabelen, $f(v_1, \dots, v_s), v_i \in \mathbb{R}^{r+1}$.

$$f(v_1, \dots, v_s) = \prod_{1 \leq i \leq s} f_i(v_i). \quad (8)$$

De dimensie van deze vectorruimte is gelijk aan $(r+1)^s$. Omdat de functies f lineair zijn kunnen we kijken naar een basis van $\mathbb{R}^{r+1}$ $B = (e_1, \dots, e_{r+1})$. Het is dan voldoende om te weten waar $f(\tilde{e}_1, \dots, \tilde{e}_{r+1})$ naar toe wordt gestuurd $\forall \tilde{e}_i, \forall i \in 1, \dots, s$. Er zijn s plaatsen op te vullen en voor elke plaats hebben we voor de $\tilde{e}_i$ keuze uit $r+1$ basiselementen. Dus de totale mogelijkheden worden dan $(r+1)^s$. Dus de dimensie van deze vectorruimte wordt $(r+1)^s$.

Een multi-lineaire functie heet symmetrisch als de waarde invariant is onder permutatie van zijn variabelen. Dat wil zeggen dat als $\rho(i), i \in 1, \dots, s$ een willekeurige permutatie is van $\{1, \dots, s\}$, dan geldt dat $f(v_i, \dots, v_s) = f(v_{\rho(1)}, \dots, v_{\rho(s)})$. Wat is dan de dimensie van vectorruimte van symmetrische multi-lineaire functies?

Omdat de functies symmetrisch zijn maakt het niet meer uit in welke volgorde de variabele zijn, want immers de waarde van f was invariant onder het nemen van permutaties. Omdat we nu weer werken met multi-lineaire functies, is het weer genoeg om te kijken naar een basis van $\mathbb{R}^{r+1}$ $B = (e_1, \dots, e_{r+1})$ en $f(\tilde{e}_1, \dots, \tilde{e}_{r+1})$. Maar omdat nu de volgorde ook niet meer uitmaakt hoeven we alleen nog maar te kijken hoe vaak een $e_i \in B$ voorkomt tussen onze variabelen in $f(\tilde{e}_1, \dots, \tilde{e}_{r+1})$.

Er zijn $r+1$ mogelijke e'_i 's en we willen kijken hoe vaak elke e_i wordt gebruikt. We kunnen het probleem dan reduceren tot het volgende probleem:

We hebben $r+1$ bakken en s ballen, op hoeveel manieren kunnen we deze s ballen verdelen over de $r+1$ bakken?

Met lemma 3.6 kan dit op $\binom{r+s}{s}$ manieren.

De dimensie van deze deelruimte is dus gelijk aan $\binom{r+s}{s}$

$\forall B_j = \{b_1^j, \dots, b_s^j\}$ definiëren we een symmetrische multi-lineaire functie:

$$f_{B_j} = f_{B_j}(v_1, \dots, v_s) = \frac{1}{s!} \sum_{\rho \in S_s} \prod_{1 \leq i \leq s} f_{b_{\rho(i)}}(v) \quad (9)$$

Dit zijn m symmetrische multi-lineaire functies in een ruimte van $\binom{r+s}{s}$ dus als we bewijzen dat deze f'_{B_j} 's lineair onafhankelijk zijn dan geldt er dat $m \leq \binom{r+s}{s}$, dan hebben we het gevraagde bewezen.

Dus we willen bewijzen dat deze f'_{B_j} 's lineair onafhankelijk zijn. Dit gaan we doen met behulp van tegenpraak.

Stel ze zijn niet lineair onafhankelijk. Als ze niet lineair onafhankelijk zijn betekent dat dat er $c_j \in \mathbb{R}$ bestaan zodanig dat $f = \sum_j c_j f_{B_j}$ is de nul-functionaal voor $c_j \neq 0, \forall i \in \{1, \dots, m\}$.

Laat t de eerste j zijn zodanig dat $c_j \neq 0$, dan willen we bepalen wat de waarde is van $f(u_t, \dots, u_t)$. We kunnen uit (9) zien dat

$$f_{B_j}(u_t, \dots, u_t) = \prod_{1 \leq i \leq s} (b_i^j, u_t) \quad (10)$$

want we kijken alleen maar naar u_t en er is maar een permutatie mogelijk, namelijk de permutatie die niks doet. dus $\frac{1}{s!} = 1$ en bij de som wordt alleen maar 1 term bekeken. Wat we dan overhouden is wat we hierboven zagen.

Voor $t < j \leq m$ geldt met de aannames uit de stelling dat $B_j \cap A_t \neq \emptyset$. Dan geldt dat $(b_i^j, u_t) = 0$ voor de bijbehorende b_i^j vanwege het feit dat u_t de eenheid-normaal vector is op V_t . Omdat nu een van de termen in het product gelijk is aan 0, geldt dat (10) gelijk is aan 0.

Als $c_j = 0$ voor $1 \leq j < t$ hebben we laten zien dat

$$c_j f_{B_j}(u_t, \dots, u_t) = 0 \quad j \neq t \quad (11)$$

Met de andere aanname uit de stelling weten we dat $A_t \cap B_t = \emptyset$. Ook liggen de punten in generale positie. Daarmee zien we dat $B_t \cap V_t = \emptyset$. en omdat nu alle punten $b_i^j \in B_j$ zeker niet in V_j liggen, geldt dat $(b_i^j, u_t) \neq 0$ Dit zorgt er voor dat

$$c_t f_{B_t}(u_t, \dots, u_t) \neq 0. \quad (12)$$

Als we dit samen nemen zien we dat

$$f(u_t, \dots, u_t) = \sum_j c_j f_{B_j}(u_t, \dots, u_t) \neq 0. \quad (13)$$

Want een van de termen in de som is niet 0. Nu hebben we een tegenspraak met wat we hebben aangenomen. Dus alle f_{B_j} zijn lineair onafhankelijk, dus geldt er dat $m \leq \binom{r+s}{s}$ en daarmee is het gevraagde bewezen. $\square$

Bewijs. Bewijs van Stelling 3.4

[2] Voor dit bewijs hebben we het volgende algoritme nodig:

Algoritme

Input: $\mathcal{A} = \langle Q, \Sigma, \delta \rangle$ (Een DFA)

Initialisatie: $w \leftarrow \lambda$ (het lege woord)

$P \leftarrow Q$

while ($|P| > 1$)

Vind een woord $v \in \Sigma^*$ van minimale lengte zodanig dat $|\delta(P, v)| < |P|$

if zo een v niet bestaat **return FALSE**

$w \leftarrow wv$

$P \leftarrow \delta(P, v)$

return w

Dit is een algoritme om een synchroniserend woord w voor de automaat $\mathcal{A}$ te vinden. In voorbeeld 2.12 zien we dat het algoritme niet een kortst synchroniserend woord maakt, want het algoritme is erg greedy en probeert zo snel mogelijk naar een kleinere verzameling te gaan, dit zorgt er echter niet voor dat het gevonden synchroniserende woord w ook het kortst is.

Als er geen woord kan worden gemaakt wordt er FALSE gereturned.

We willen bewijzen dat geldt dat $|w| \leq \frac{n^3 - n}{6}$.

Als $|Q| = n$ dan wordt het algoritme maximaal $n - 1$ keer uitgevoerd.

Om de lengte van w te berekenen gaan we schatten wat de lengte van het woord v wordt dat geproduceerd wordt door het algoritme.

Beschouw een algemene stap in het algoritme waarvoor geldt dat $|P| = k > 1$. Laat $v = a_1 a_2 \dots a_l$ met $a_i \in \Sigma^*$, dan willen we dus l gaan afschatten, omdat we dan een schatting voor de lengte van v als $|P| = k > 1$ hebben.

Als we hierna $\sum_{k=2}^n l$ berekenen hebben we de lengte van het synchroniserende woord w , dat wordt geproduceerd door het algoritme.

Maar eerst gaan we de waarde van l in de stap $|P| = k > 1$ schatten.

We hebben $|P| = k > 1$ en $v = a_1 a_2 \dots a_l$ met $a_i \in \Sigma^*$ dan geldt dat:

$P_1 = P, P_2 = \delta(P_1, a_1), \dots, P_l = \delta(P_{l-1}, a_{l-1})$ en $|P_i| = k$ voor $i = 1, \dots, l$.

Dit is omdat $v = a_1 a_2 \dots a_l$ synchroniserend is, als het woord korter is dan dit dan is het nog niet synchroniserend en dus worden nog geen twee toestanden naar dezelfde toestand gestuurd door δ . Stel dat er voor een van de p'_i s geldt dat $|P_i| < k$ dan is er dus een korter woord $\tilde{v}$ met de eigenschap $|\delta(p, \tilde{v})| < |P|$, maar dat was niet het geval omdat v de kortste was.

We hebben dus een rij $P_1, \dots, P_l$ met $|P_i| = k$ voor $i = 1, \dots, l$

Omdat v een minimaal synchroniserend woord is geldt dat $|\delta(P, v)| < |P|$. Dat wil zeggen dat er minstens twee toestanden $q_l, q'_l \in P_l$ zijn zodanig dat $\delta(q_l, a_l) = \delta(q'_l, a_l)$.

We definiëren nu een twee-element verzameling $R_i = \{q_i, q'_i\}, i = 1, \dots, l$ met de eigenschappen dat

- $\delta(q_i, a_i) = q_{i+1}$
- $\delta(q'_i, a_i) = q'_{i+1}$

Voor deze verzameling geldt het volgende:

$R_i \subseteq P_i$ en $R_i \not\subseteq P_j$ als $j < i$, want v is een woord van minimale lengte met $|\delta(P, v)| < |P|$.

We definiëren nu het volgende $\tilde{P}_i = Q \setminus P_i$, dan $|\tilde{P}_i| = n - k$ voor $i = 1, \dots, l$ en $\tilde{P}_i \cap R_i = \emptyset$ en $\tilde{P}_j \cap R_i \neq \emptyset$ voor $i > j$. Neem nu twee families van verzamelingen: $\mathcal{A} = \{\tilde{P}_1, \dots, \tilde{P}_l\}$ familie van $(n - k)$ -verzamelingen en $\mathcal{B} = \{R_1, \dots, R_l\}$ familie van 2-verzamelingen. Deze twee verzamelingen voldoen aan de eisen van de stelling van Frankl. Dus geldt er dat

$$l \leq \binom{n - k + 2}{2}$$

Als we nu de som nemen over $k = 2, \dots, n$.

$$\sum_{k=2}^n \binom{n - k + 2}{2} = \frac{n^3 - n}{6}.$$

Hiermee is het gevraagde bewezen. □

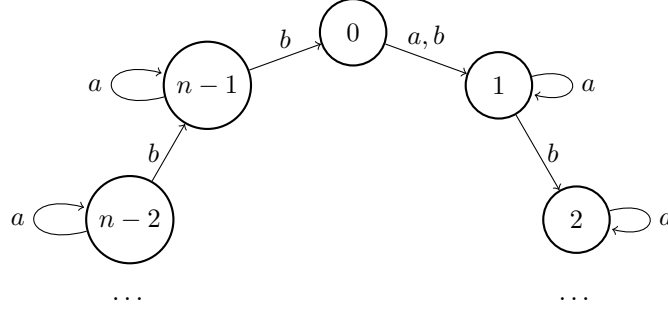
3.2 Černý automaten

Het is dus tot op de dag van vandaag nog niet gelukt om het vermoeden van Černý te bewijzen. Černý kwam in 1964 met een constructie van de automaat weergegeven in figuur 5. We noemen deze automaten Černý automaten, $\mathcal{C}_n$.

Definieer s_i als de i^e toestand, met $i = 0, \dots, n - 1$ dan wordt de transitie functie als volgt gedefinieerd:

$$\begin{aligned} \delta : Q \times \Sigma &\longrightarrow Q : \\ \delta(s_i, a) &= s_i \quad \text{voor } 1 \leq i \leq n - 1 \\ \delta(s_i, b) &= s_{i+1} \quad \text{voor } 1 \leq i \leq n - 1 \\ \delta(s_0, a) &= s_1 \\ \delta(s_0, b) &= s_1 \end{aligned} \tag{14}$$

Hij bewees dat deze automaat synchroniserend was en het kortste synchroniserende woord dat hij vond was $(ab^{n-1})^{n-2}a$. De lengte van dit woord is gelijk aan $(1 + (n - 1))(n - 2) + 1 = (n - 1)^2$. We zien dus dat deze speciale klasse van synchroniserende automaten de grens van het Černý vermoeden bereikt. Het bewijs hiervoor gaat als volgt .



Figuur 5: Černý automaat $\mathcal{C}_n$

Stelling 3.7. Voor $\mathcal{C}_n, n \in \mathbb{N}$, geldt dat de lengte van het kortste synchroniserende woord gelijk is aan $(n-1)^2$

Bewijs. [4] Bekijk de transitie functie van $\mathcal{C}_n$ met $n \in \mathbb{N}$ toestanden, gegeven in vergelijking (14). We definiëren w als een woord, dan is w_i de prefix van w van lengte i .

We definiëren $l(i)$ als de lengte van het kortste interval die alle toestanden van $\delta(Q, w_i)$ bevat. Waar $Q = \{s_0, \dots, s_{n-1}\}$.

We definiëren $t(j)$ als de kleinste i zodanig dat $l(i) \leq j$.

Als we bewijzen dat $\forall j < n-1$ geldt dat $t(j) \geq t(j+1) + n$. Dat wil zeggen dat we om het interval de volgende stap korter te maken, we minimaal n stapjes (letters) nodig hebben.

Als we het bovenstaande hebben bewezen zijn we klaar, want het interval moet $n-2$ korter worden gemaakt. Als elk van hen n letters kost om het interval 1 kleiner te maken, hebben we al $n(n-2)$ letters nodig. We hebben ook nog een letter nodig om de $l(t)$ te reduceren van n naar $n-1$. We hebben dan dus minimaal $n(n-2) + 1 = (n-1)^2$ letters nodig, en daarmee is het gevraagde bewezen.

We gaan bewijzen dat $\forall j < n-1$ geldt dat $t(j) \geq t(j+1) + n$. Als $j \neq 1$ en $\delta(S, w_i) \subset [s_j, s_k]$ dan zijn er twee gevallen:

- De i^e letter was een a dan weten we dat $\delta(S, w_{i-1}) \subset [s_j, s_k]$
- De i^e letter was een b dan weten we dat $\delta(S, w_{i-1}) \subset [s_{j-1}, s_{k-1}]$

We kunnen hieruit zien dat de lengte van het interval niet verandert. Daar kunnen we het volgende weer uit concluderen, $\delta(S, w_{i-j+1}) \subset I$ waar $|I| = k - j + 1$. Dit kunnen we bewijzen met inductie.

Inductie naar $j-1$ We mogen aannemen dat $\delta(S, w_i) \subset [s_j, s_k]$.

Stel $j = 2$ dan moeten we bewijzen dat $\delta(S, w_{i-1}) \subset I$ met $|I| = k - j + 1$. Dit klopt, want als de i^e letter een a is dan was $\delta(S, w_{i-1}) \subset [s_j, s_k]$ en dat is inderdaad ook een interval van lengte $k - j + 1$.

Als de i^e letter een b was, dan moest gelden dat $\delta(S, w_{i-1}) \subset [s_{j-1}, s_{k-1}]$. De lengte van dit interval is ook $k - j + 1$, dus het klopt voor $j = 2$.

Stel nu dat het klopt voor $j-1$, dan willen we bewijzen dat het klopt voor j . Met de inductie aanname weten we dat $\delta(S, w_{i-j+2}) \subset I$ met $|I| = k - j + 1$. We willen bewijzen dat $\delta(S, w_{i-j+1}) \subset I$. Als we dan als $(i-j+2)^e$ letter een a deden dan was het interval het zelfde bij $\delta(S, w_{i-j+1}) \subset I$. Dus ook hier $|I| = k - j + 1$. En als we als $(i-j+2)^e$ letter een b hadden gekozen, dan was het interval verschoven met een stapje naar rechts. Maar dan blijft de grootte van het interval gelijk aan $|I| = k - j + 1$. Dus het klopt ook voor j . Dus het gevraagde is bewezen.

We merken nu op dat als de i^e input letter een b is dan geldt er $l(i) = l(i-1)$, want we kunnen de lengte van het interval nog niet inkorten.

Dus $\forall j$ op de $t(j)^e$ positie moet een a worden gedaan, want dan wordt het interval ingekort en dat is wat we willen, want we willen dat het interval zodanig korter wordt gemaakt dat we lengte 1 krijgen, omdat we dan een synchroniserend woord hebben.

Ook moet er gelden dat $\delta(S, w_{t(j)-1}) \subset [s_n, s_j]$, want stel het is niet zo en we doen de als inputletter de

letter a , dan wordt het interval alsnog niet ingekort.

Met de bewezen inductie geldt er dat $\delta(S, w_{t(j)-n}) \subset [s_n, s_j]$ ($i = t(j) - 1, k = j, j = n$). De lengte van het interval wordt dan $j + 1$. We houden de volgende vergelijkingen over:

Er moest sowieso met de definities gelden dat:

$$|\delta(S, w_{t(j)-1})| > j$$

$$|\delta(S, w_{t(j)})| \leq j$$

Met wat we hierboven hebben verkregen geldt er dat:

$$|\delta(S, w_{t(j)-1})| = j + 1$$

$$|\delta(S, w_{t(j)})| = j$$

Alles samen genomen geldt er nu dat:

$$t(j) - 1 - t(j + 1) \leq n - 1$$

$$\Rightarrow t(j) \geq t(j + 1) + n$$

□

Dit blijkt tevens het slechtst bekende geval te zijn. Dus we hebben nog geen automaat op n toestanden gevonden die een langst kortste woord heeft. En die voldoet aan het Černý vermoeden.

Met dit bewijs is ook de ondergrens voor $C(n)$ bewezen want we hebben een automaat gevonden op n toestanden met een langst bekend kortst synchroniserend woord. Het maximum kan dus niet kleiner zijn dan $(n - 1)^2$.

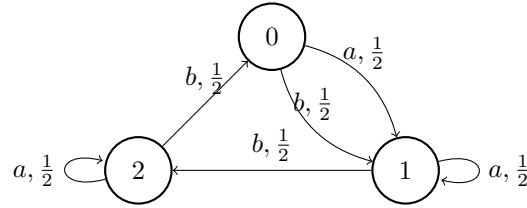
4 Random input

In dit hoofdstuk zijn we nog steeds geïnteresseerd in de lengte van synchroniserende woorden, maar er is een verschil in de input van de DFA's. Waar we in de vorige hoofdstukken keken naar automaten waar de input vast stond, gaan we nu kijken naar de zelfde soort automaten, alleen maken we de input random. Dat wil zeggen dat nog niet vast staat of we de letter a toepassen of juist de letter b . Met kans p passen we letter a toe en met kans $1 - p$ passen we de letter b toe.

De synchroniserende woorden die voorheen werkten, werken nu nog steeds alleen kan het veel langer duren voordat een automaat wordt gesynchroniseerd. We hebben namelijk geen invloed op de keuze van de letter die wordt genomen.

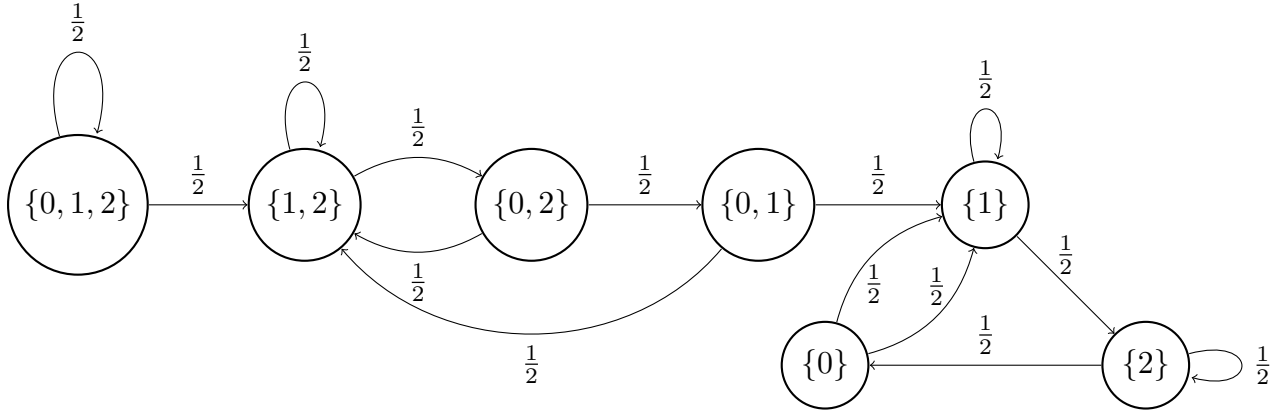
Wat wordt dan de verwachte lengte van een woord tot we een automaat gesynchroniseerd hebben? We zijn benieuwd naar de verwachte lengte van de synchroniserende woorden van deze "nieuwe" soort automaten.

Voor het gemak bekijken wij de automaten waarbij $|\Sigma| = 2$, dus bijvoorbeeld $\Sigma = \{a, b\}$ en we zeggen dat $\mathbb{P}(a) = \frac{1}{2} = \mathbb{P}(b)$. Bijvoorbeeld als we kijken naar de volgende automaat, de kans op elke pijl is $\frac{1}{2}$.



Figuur 6: Automaat met random input

Bij deze random automaat hoort de volgende powerautomaat met kansen.



Figuur 7: Powerautomaat van random automaat uit figuur 6

4.1 Markovketens

Als we kijken naar de powerautomaten van random automaten, in bijvoorbeeld 7, dan herkennen we er een markov keten in. Om te definiëren wat een markovketen is hebben we eerst een aantal definities nodig [5]:

Definitie 4.1. We zeggen dat $\lambda = (\lambda_i : i \in S)$ een maat is op S als $0 \leq \lambda_i \leq \infty \forall i \in S$.

Als er geldt dat $\sum_{i \in S} \lambda_i = 1$ dan heet λ een verdeling.

Stel we nemen aan dat $\lambda_i = \mathbb{P}(X = i)$ met X een random variabele met waardes in S . Dan definieert λ een verdeling, de verdeling van X .

Definitie 4.2. Laat $P = (p_{ij} : i, j \in S)$ een matrix, dan heet P stochastisch als elke rij van $(p_{ij} : j \in S)$ een verdeling is.

We zullen de regels voor een markov keten definiëren aan de hand van de corresponderende matrix P die hierboven is gedefinieerd.

Definitie 4.3. Een stochastisch proces $\{X_n | n \geq 0, n \in S\}$ met toestand-ruimte S heet een markov keten met begin verdeling λ en transitie matrix P als:

1. X_0 heeft verdeling λ
2. voor $n \geq 0$ conditioneel op $X_n = i, X_{n+1}$ heeft verdeling $(p_{ij} | j \in S)$ en is onafhankelijk van $X_0, \dots, X_n$

Wat explicieter geldt dat

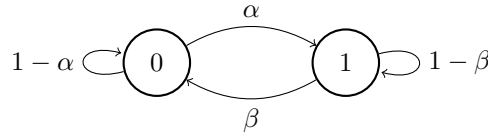
1. $\mathbb{P}(X_0 = i_0) = \lambda_{i_0}$
2. $\mathbb{P}(X_{n+1} = i_{n+1} | X_0 = i_0, \dots, X_n = i_n) = p_{i_n i_{n+1}}$

waarbij $\lambda_i = \mathbb{P}(X = i)$

We zeggen dat $(X_n)_{n \geq 0}$ is Markov(λ, P)

Een voorbeeld van een markov keten is het volgende:

Voorbeeld 4.4. $\begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$ Met bijbehorende markov keten



Figuur 8: Markov keten

4.2 Berekening verwachte lengte van een synchroniserend woord

Voor deze sectie hebben we nieuwe definities nodig:

Definitie 4.5. We definiëren de verzameling A als $A = \{S \subset Q | |S| = 1\}$. A is de verzameling van de een-punt-verzamelingen uit $\mathcal{P}(Q)$.

Definitie 4.6. $K_S^A = \mathbb{E}(\text{aantal letters tot je bij } A \text{ bent beginnend in } S)$

We willen graag weten wat de verwachte lengte van een synchroniserend woord voor een automaat $\mathcal{A}$ is. Kort gezegd we willen berekenen wat K_Q^A is, want dit geeft precies de verwachte lengte van Q naar een een-punt-verzameling in een powerautomaat. In de tweede sectie hebben we gezien dat een synchroniserend woord voor automaat $\mathcal{A}$ een pad is van Q naar een een-punt-verzameling in de powerautomaat van automaat $\mathcal{A}$. Als we dus de K_Q^A berekenen weten we de verwachte lengte van zo een pad en hebben we precies de verwachte lengte van een synchroniserend woord voor de automaat $\mathcal{A}$. Omdat de power-automaten in het geval van random input markovketens worden kunnen we met behulp van het volgende stelsel K_Q^A berekenen.

Stelling 4.7. [5] De vector van gemiddelde raaktijden $k^A = (k_i^A : i \in S)$ is de minimale niet-negatieve oplossing van het systeem van lineaire vergelijkingen.

$$\begin{cases} K_i^A = 0 & \text{als } i \in A \\ K_i^A = 1 + \sum_{j \notin A} p_{ij} K_j^A & \text{als } i \notin A \end{cases} \quad (15)$$

Voorbeeld 4.8. Als we kijken naar de automaat in figuur 7 dan kunnen we K_Q^A als volgt uitrekenen met behulp van het stelsel in vergelijking (15):

$$\begin{cases} K_{\{0,1,2\}}^A = 1 + \frac{1}{2}K_{\{0,1,2\}}^A + \frac{1}{2}K_{\{1,2\}}^A \\ K_{\{1,2\}}^A = 1 + \frac{1}{2}K_{\{1,2\}}^A + \frac{1}{2}K_{\{0,2\}}^A \\ K_{\{0,2\}}^A = 1 + \frac{1}{2}K_{\{1,2\}}^A + \frac{1}{2}K_{\{0,1\}}^A \\ K_{\{0,1\}}^A = 1 + \frac{1}{2}K_{\{1,2\}}^A + \frac{1}{2}K_{\{1\}}^A \end{cases}$$

Dit resulteert in:

$$\Rightarrow K_{\{0,1\}}^A = 1 + \frac{1}{2}K_{\{1,2\}}^A$$

$$\Rightarrow K_{\{0,2\}}^A = \frac{3}{2} + \frac{3}{4}K_{\{1,2\}}^A$$

$$\Rightarrow K_{\{1,2\}}^A = 14$$

$$\Rightarrow K_{\{0,1,2\}}^A = 16$$

Er is nog een manier om uit te rekenen wat de verwachte lengte van een synchroniserend woord voor automaat $\mathcal{A}$ is. Hier hebben we nog een extra definitie voor nodig.

Definitie 4.9. Laat m_{ij} het verwachte aantal keer dat we van toestand i naar toestand j gaan. Laat p_{ij} de kans om van toestand i naar toestand j te gaan.

We gaan K_S^A berekenen met behulp van het verwachte aantal keer dat een pijl van de ene naar de andere toestand gebruikt wordt. We gaan een stelsel opstellen aan de hand van de volgende regels:

Regel 1: Het verwachte aantal keer dat we de pijl van een toestand buiten A naar een toestand binnen A gebruiken is 1.

Regel 2: Het verwachte aantal keer dat een pijl van toestand i naar toestand j en k wordt gebruikt is afhankelijk van de kans dat een pijl wordt gebruikt. Er geldt dat $p_{ij}m_{ik} = p_{ik}m_{ij}$

Regel 3: De som van het aantal verwachte keren dat een pijl in een toestand gaat is gelijk aan het aantal verwachte keren dat een pijl uit dezelfde toestand gaat. Mits voor toestand i geldt dat $i \neq S, i \notin A$

Regel 4: Als je geïnteresseerd bent in de verwachte lengte van een woord om te synchroniseren, dan is de som van het verwachte aantal keer dat uit de toestand behorend bij S gaat één meer dan het aantal verwachte keren dat men in die toestand komt.

Als we deze regels op de juiste manier toepassen kunnen we het verwachte aantal keer dat een pijl gebruikt wordt berekenen.

Deze regels vertalen we naar het volgende stelsel.

Er geldt het volgende stelsel voor $i, j \in \mathcal{P}(Q)$:

$$\begin{cases} \sum_i m_{iA} = 1 & \text{voor } i \notin A \\ p_{ik}m_{ij} = p_{ij}m_{ik} & \text{voor vaste } i \\ \sum_i m_{ij} = \sum_i m_{ji} & \text{als } j \neq Q \\ 1 + \sum_i m_{ij} = \sum_i m_{ji} & \text{als } j = Q \\ m_{ij} = 0 & \text{voor } i, j \in A \end{cases} \quad (16)$$

Er geldt dat

$$K_Q^A = \sum_{i,j} m_{ij} \quad (17)$$

Want m_{ij} staat voor het aantal verwachte aantal keren dat de pijl van toestand i naar toestand j wordt gebruikt. Als we dan al deze verwachtingen bij elkaar optellen dan hebben we het verwachte aantal keer dat we de pijlen van de hele powerautomaat gebruiken en hiermee hebben we het verwachte aantal letters die we nodig hebben om van Q naar A te komen. We hebben dus twee manieren om de verwachting K_Q^A uit te rekenen.

Voorbeeld 4.10. We gaan weer kijken naar de automaat gegeven in figuur 7. Nu passen we het stelsel van vergelijking (16) toe. Voor het gemak schrijven we de toestanden als volgt: $\{0, 1, 2\} = a, \{1, 2\} = b, \{0, 2\} =$

$c, \{0, 1\} = d, \{1\} = e.$

Als we stelsel 2 toepassen krijgen we de volgende vergelijkingen:

$$\begin{cases} m_{de} = 1 \\ \frac{1}{2}m_{de} = \frac{1}{2}m_{db} \\ m_{cd} = m_{db} + m_{de} \\ \frac{1}{2}m_{cd} = \frac{1}{2}m_{cb} \\ m_{bc} = m_{cd} + m_{cb} \\ \frac{1}{2}m_{bc} = \frac{1}{2}m_{bb} \\ m_{ab} + m_{cb} + m_{db} + m_{bb} = m_{bb} + m_{bc} \\ 1 + m_{aa} = m_{aa} + m_{ab} \\ \frac{1}{2}m_{aa} = \frac{1}{2}m_{ab} \end{cases}$$

Dit resulteert in :

$$m_{de} = 1 = m_{db}$$

$$m_{cd} = 2 = m_{cb}$$

$$m_{bc} = 4 = m_{bb}$$

$$m_{ab} = 1 = m_{aa}$$

$$\text{Dan } K_Q^A = 1 + 1 + 2 + 2 + 4 + 4 + 1 + 1 = 16$$

We zien dat deze stelsel inderdaad dezelfde uitkomst hebben. Ook zien we dat op het eerste gezicht het stelsel in vergelijking (15) makkelijker lijkt dan het stelsel in vergelijking (16) . Maar in de praktijk is het stelsel in vergelijking (16) veel makkelijker en sneller uit te rekenen.

4.3 Synchronisatie Černý automaat

In deze deel-sectie kijken we naar een speciale klasse van automaten, namelijk de Černý automaten $\mathcal{C}_n$ met n toestanden. Deze klasse van automaten kennen we al uit de vorige sectie. In het geval met deterministische input is de Černý automaat het slechts bekende geval, wat wil zeggen de automaat op n toestanden met het langste kortste synchroniserende woord. Doet deze automaat er ook het langst over in het geval van random input? Wat is de K_Q^A van deze klasse van automaten?

De verwachte lengtes van het kortste synchroniserende woord voor $\mathcal{C}_n$ met $n = 2, 3, 4$ staan weergegeven in de onderstaande tabel. Deze zijn berekend met de eerste en tweede methode. We zien dat de lengte relatief snel om hoog gaat.

n	2	3	4
K_Q^A	2	16	50

We zijn nu geïnteresseerd in een afchatting die ons iets vertelt over de verwachte lengte van het kortste synchroniserende woord.

Stel we bekijken de eerder genoemde automaten $\mathcal{C}_n$, met $n \in \mathbb{N}$ en we gooien een munt op, daarbij doen we de letter a als er kop tevoorschijn komt en nemen we de letter b als er munt te voorschijn komt. Beide routes worden dus met evenveel kans genomen. We zijn nu benieuwd naar de verwachte lengte van een synchroniserend woord.

Om hier een goede afchatting voor te maken, gaan we eerst kijken naar de verwachte lengte van een woord om een willekeurig paar uit $\mathcal{C}_n$ te synchroniseren.

Als eerste maken we de parenautomaat, deze heeft de volgende definitie.

Definitie 4.11. Een parenautomaat van een automaat $\mathcal{A} = (Q, \Sigma, \delta)$ is een 3-tupel $(\tilde{Q}, \tilde{\Sigma}, \tilde{\delta})$ zodanig dat:

$\tilde{Q} = \{\{p, q\} | p, q \in Q, p \neq q\}$

$\tilde{\Sigma}: \Sigma$, het alfabet van de automaat $\mathcal{A}$.

$\tilde{\delta}: \text{De transitie functie } \tilde{\delta}: Q^2 \times \Sigma^* \longrightarrow Q$

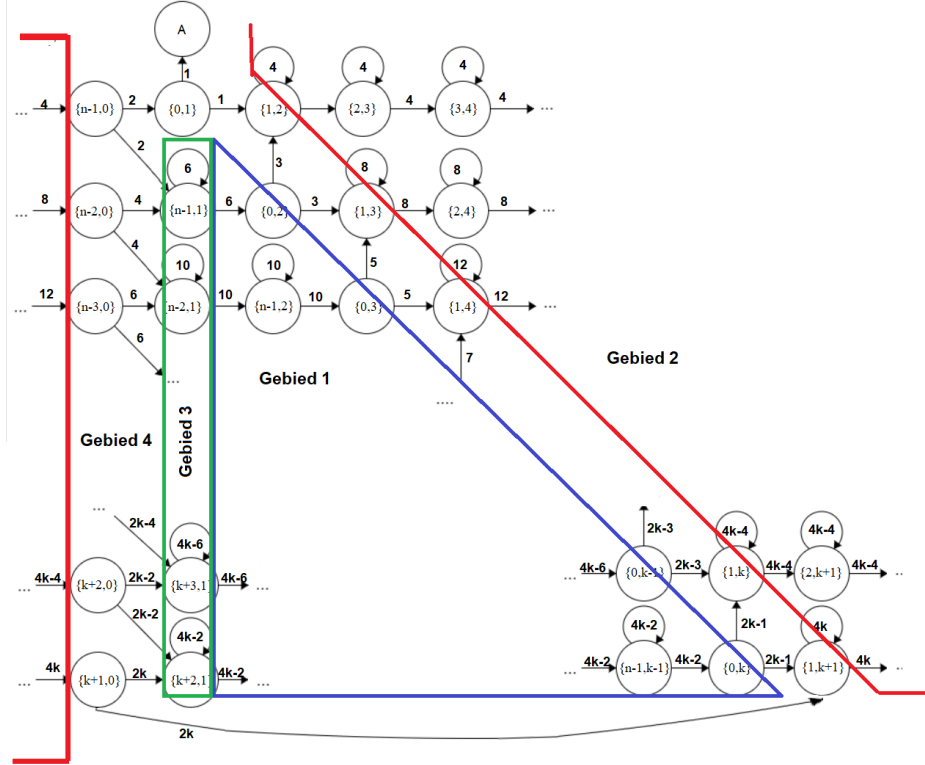
$\tilde{\delta}(\{q_1, q_2\}, w) = \{\delta(q_1, w), \delta(q_2, w)\}$ met $w \in \Sigma^*$, $q_1, q_2 \in Q$ en δ de transitie functie van $\mathcal{A}$

Opmerking: Wij zijn geïnteresseerd in synchronisatie dus we moeten een paar uiteindelijk naar een een-punt-verzameling brengen. In de parenautomaat maken we daarom een toestand A gedefinieerd in definitie 4.5. Deze toestand representeert alle een-punt-verzamelingen.

Stelling 4.12. De verwachte lengte van een synchroniserend woord voor het paar $\{1, k+1\}$ in de Černý automaat is gelijk aan:

$$\begin{cases} n^3 - \frac{3}{2}n^2 + \frac{1}{2} & \text{als } n = 2k+1, k \in \mathbb{N} \\ n^3 + \frac{5}{6}n^2 - \frac{4}{3}n + 2 & \text{als } n = 2k, k \in \mathbb{N} \end{cases} \quad (18)$$

Figuur 9: Parenautomaat met verwachte aantal transities



Bewijs. [6] Als eerst maken we de parenautomaat, zie figuur 9, Nu zijn we geïnteresseerd in hoeveel letters we verwachten nodig te hebben om van de toestand $\{1, k+1\}$ naar A te komen, want dan is dit paar gesynchroniseerd.

Om dit te kunnen berekenen moeten we weten hoe vaak we verwachten dat we de pijl van de ene toestand naar de andere toestand gaan gebruiken. Om dit te berekenen maken gebruik van de volgende regels:

Regel 1: Het verwachte aantal keer dat we de pijl van $\{0, 1\}$ naar A gebruiken is 1.

Regel 2: Het verwachte aantal keer dat een pijl wordt gebruikt, is voor beide pijlen uit een toestand gelijk, immers de kans op de letter a is even groot als de kans op de letter b .

Regel3: De som van het aantal verwachte keren dat een pijl in een toestand gaat is gelijk aan het aantal verwachte keren dat een pijl uit dezelfde toestand gaat, mits je niet in die toestand begint.

Regel 4: Als je geïnteresseerd bent in de verwachte lengte van een woord om het paar $\{u, v\}$ te synchroniseren, dan is de som van het verwachte aantal keer dat uit de toestand behorend bij het paar $\{u, v\}$ gaat één meer dan het aantal verwachte keren dat je in die toestand komt.

Als we deze regels op de juiste manier toepassen kunnen we het verwachten aantal keer dat een pijl gebruikt wordt berekenen.

Elke keer dat we een pijl verwachten te gebruiken hebben we een letter nodig. Dus als we de som van alle verwachte aantallen dat een pijl gebruikt wordt bij elkaar optellen, dan hebben we het verwachte aantal letters dat we nodig hebben om het paar $\{1, k+1\}$ te synchroniseren. Op deze manier hebben we de verwachte lengte van het synchroniserende woord voor dat paar.

Om de som van alle verwachte aantallen per pijl bij elkaar op te tellen, delen we de parenautomaat op in

vier gebieden. In elk van deze gebieden is het berekenen van de som relatief gemakkelijk en als we dan later de gebieden weer bij elkaar optellen hebben we het totaal.

Opmerking: We maken gebruik van het feit dat n oneven of even is. Dat wil zeggen dat we kunnen schrijven als $n = 2k + 1$ of $n = 2k$.

We beginnen in het geval dat n oneven is en er geldt dat: $n = 2k + 1$

gebied 1:

$$\begin{aligned} & 1 \cdot 6 + 3 \cdot 10 + \dots + (2(k-1) - 1) \cdot (4k + 2) \\ &= \sum_{i=1}^{k-1} (2i-1)(4i+2) \\ &= \sum_{i=1}^{k-1} 8i^2 - 2 \\ &= 8 \sum_{i=1}^{k-1} i^2 - 2(k-1) \\ &= \frac{8}{3}k^3 - 4k^2 - \frac{2}{3}k + 2 \\ &= \frac{1}{3}n^3 - 2n^2 + \frac{8}{3}n + 1 \end{aligned}$$

gebied 2:

$$\begin{aligned} & (2n-4) \cdot 4 + (2n-6) \cdot 8 + \dots + (2n-2(k+1)) \cdot 4k \\ &= \sum_{i=1}^k 4i(2n-2-2i) \\ &= \sum_{i=1}^k 8ni - 8i - 8i^2 \\ &= 8n \sum_{i=1}^k i - 8 \sum_{i=1}^k i - 8 \sum_{i=1}^k i^2 \\ &= \frac{8}{3}n(k^2+k) - 4(k^2+k) \\ &= \frac{2}{3}n^3 - n^2 - \frac{2}{3}n + 1 \end{aligned}$$

gebied 3:

$$\begin{aligned} & 6 + 10 + \dots + 4k - 2 \\ &= \sum_{i=1}^{k-1} 2 + 4i \\ &= 2(k-1) + 4 \sum_{i=1}^{k-1} i \\ &= 2k^2 - 2 \\ &= \frac{1}{2}(n^2 - 2n - 3) \end{aligned}$$

gebied 4:

$$\begin{aligned} & 2(1 + 2 + \dots + 2k) \\ &= \sum_{i=1}^{2k} 2i \\ &= 2 \sum_{i=1}^{2k} i \\ &= 2 \frac{(2k+1)(2k)}{2} \\ &= (2k+1)(2k) \\ &= n(n-1) \\ &= n^2 - n \end{aligned}$$

Totaal

$$= n^3 - \frac{3}{2}n^2 + \frac{1}{2}$$

Het geval dat n is even, $n = 2k$, gaat analoog.

Totaal

$$= n^3 + \frac{5}{6}n^2 - \frac{4}{3}n + 2$$

□

Definitie 4.13. Definieer de onderlinge afstand voor de toestanden p, q als volgt.

Stel $q > p$ dan

$$\text{dist}(p, q) = \min\{q - p, p - q + n\}$$

Lemma 4.14. Het paar $\{1, k+1\}$ heeft de maximale verwachte lengte van het synchroniserende woord.

Bewijs. Het paar $\{1, k+1\}$ heeft de maximale onderlinge afstand, namelijk k . We moeten de onderlinge afstand met k verkleinen om het paar te synchroniseren.

Er is in de rij met toestanden met onderlinge afstand k een paar waarbij we in één stap de onderlinge afstand met 1 kunnen verlagen. Dat is de toestand behorend bij het paar $\{0, k\}$; het paar $\{1, k+1\}$ doet er het langst over om naar het paar $\{0, k\}$ te komen. Daarom is dit het paar dat de maximale verwachte lengte heeft. □

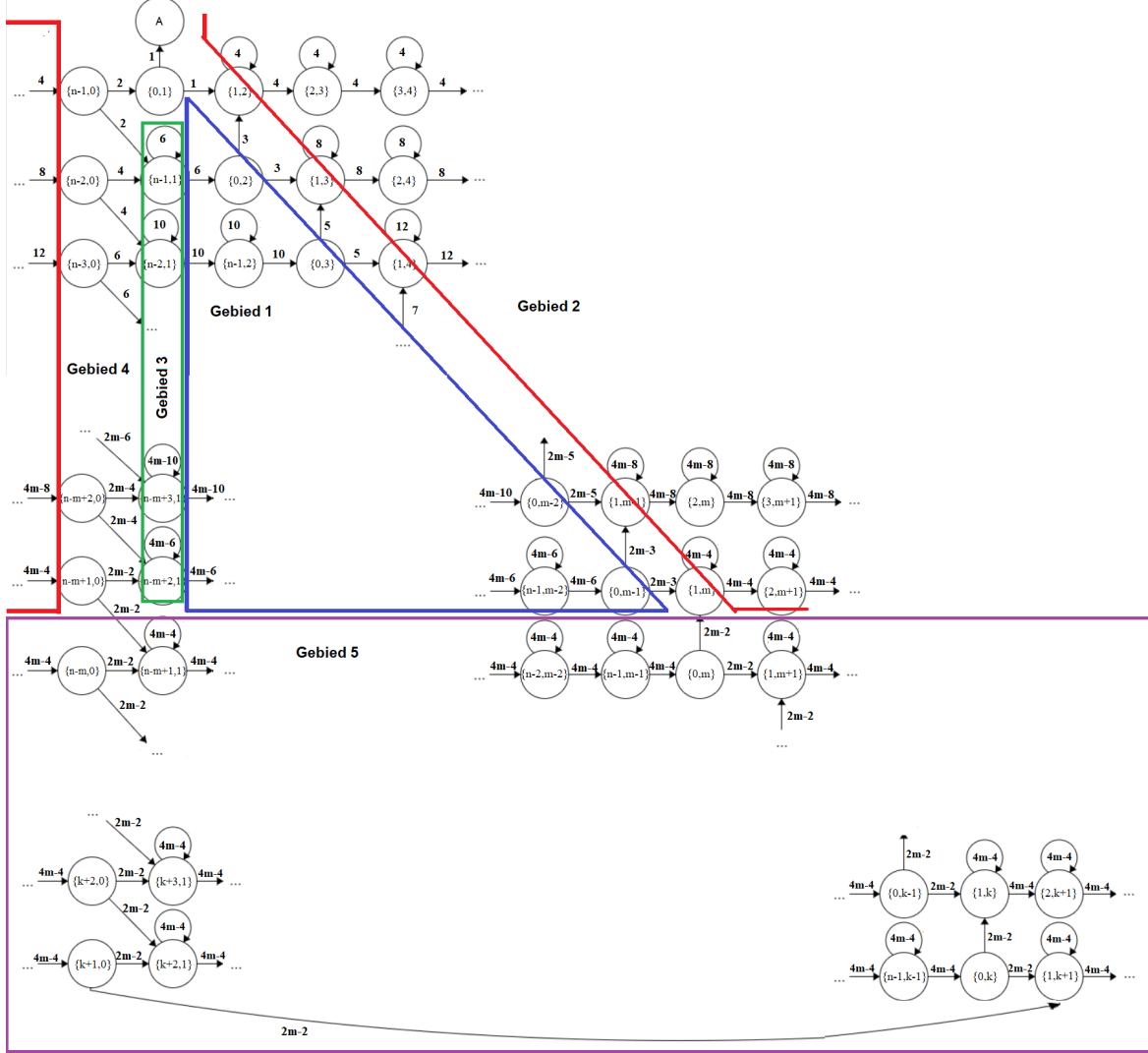
We zijn niet alleen geïnteresseerd in paar dat er het langst over doet om te synchroniseren we willen ook van

een willekeurige afstand $m - 1 < k + 1$ weten wat de verwachte lengte van het woord is om het paar $\{1, m\}$ te synchroniseren.

Stelling 4.15. *De verwachte lengte voor het synchroniserende woord voor het paar $\{1, m\}$ in de Černý automaat is gelijk aan:*

$$\begin{cases} 4(m-1)n^2 - 4m(m-1)n + m(2m-1) + 2 & \text{als } n = 2k+1, k \in \mathbb{N} \\ 4(m-1)n^2 - 4(m^2 - 2m + 1)n + 2(m^2 - 4m) - 6 & \text{als } n = 2k, k \in \mathbb{N} \end{cases} \quad (19)$$

Figuur 10: Paren automaat



Bewijs. [6] Dit bewijs gaat met behulp van figuur 10 analoog aan het bewijs van stelling 4.12.

$n = 2k + 1$

Gebied 1:

$$\begin{aligned} & \sum_{i=1}^{m-2} (2i-1)(4i+2) \\ &= \sum_{i=1}^{m-2} 8i^2 - 2 \\ &= 8 \sum_{i=1}^{m-2} i^2 - 2(m-2) \\ &= \frac{8}{3}m^3 - 12m^2 + \frac{46}{3}m - 4 \end{aligned}$$

Gebied 2

$$\begin{aligned}
& \sum_{i=1}^{m-1} 4i(2n-2-2i) \\
&= \sum_{i=1}^{m-1} 8ni - 8i - 8i^2 \\
&= 8n \sum_{i=1}^{m-1} i - 8 \sum_{i=1}^{m-1} i - \sum_{i=1}^{m-1} i^2 \\
&= 4nm^2 - 4nm - \frac{8}{3}m^3 + \frac{8}{3}m
\end{aligned}$$

Gebied 3:

$$\begin{aligned}
& \sum_{i=1}^{m-2} 2 + 4i \\
&= 2(m-2) + 4 \sum_{i=1}^{m-2} i \\
&= 2m^2 - 4m
\end{aligned}$$

Gebied 4:

$$\begin{aligned}
& 2(1 + 2 + \dots + 2m - 2) \\
&= 2 \sum_{i=1}^{2m-2} i \\
&= 2 \frac{(2m-2)(2m-1)}{2} \\
&= 4m^2 - 6m + 2
\end{aligned}$$

Gebied 5 kunnen we weer opdelen in 2 stukken, namelijk de stukken van $2m-2$ en die van $4m-4$:

2m-2:

$$\begin{aligned}
& 4 * (k-m+1)(2m-2) \\
&= -8m^2 + 4mn + 12m - 4n - 4
\end{aligned}$$

4m-4

$$\begin{aligned}
& (k-m+1)(2n-4)(4m-4) \\
&= -8m^2n + 16m^2 + 4mn^2 + 4mn - 24m - 4n^2 + 4n + 8
\end{aligned}$$

Totaal:

$$= 4(m-1)n^2 - 4m(m-1)n + m(2m-1) + 2$$

$n = 2k$ Het geval dat n is even gaat analoog aan het geval dat n oneven is.

Totaal:

$$= 4(m-1)n^2 - 4(m^2 - 2m + 1)n + 2(m^2 - 4m) - 6$$

□

Lemma 4.16. *Voor alle paren met onderlinge afstand $m-1$ heeft het paar $\{1, m\}$ de maximale verwachte lengte van het synchroniserende woord.*

Bewijs. Dit kunnen we eenvoudig laten zien als we kijken naar de parenautomaat in figuur 10, want als we kijken naar de rij waarbij de onderlinge afstand steeds $m-1$ is, dan kunnen we bij het paar $\{0, m-1\}$ naar het paar $\{1, m-1\}$ met onderlinge afstand $m-2$ of we kunnen naar het paar $\{1, m\}$ met onderlinge afstand $m-1$.

De enige manier om de onderlinge afstand te verkleinen is via het paar $\{0, m-1\}$. Bij elk ander paar in deze rij kom je of in een paar met grotere onderlinge afstand, of een gelijke onderlinge afstand en het paar dat er het langst over doet om weer naar $\{0, m-1\}$ te komen is het paar $\{1, m\}$. Dit paar heeft dus de maximale verwachte lengte van het synchroniserende woord. □

Stelling 4.17. *Zij r_n de verwachte lengte van het synchroniserende woord voor C_n dan geldt er $\forall n \in \mathbb{N}$ dat*

$$n^3 - \frac{3}{2}n^2 + \frac{1}{2} \leq r_n \leq 4n^3 \log(n) + \mathcal{O}(n^3) \quad (20)$$

Bewijs. $n^3 - \frac{3}{2}n^2 + \frac{1}{2} \leq r_n$

Om de hele automaat $\mathcal{A}$ te synchroniseren moet elk paar dat gemaakt kan worden met behulp van de toestanden uit $\mathcal{A}$ worden gesynchroniseerd. Ook het slechtste paar. Uit lemma 4.14 weten we dat $\{1, k+1\}$ er het langst over doet om te synchroniseren. Uit stelling 4.12 weten we dat de verwachte lengte van het synchroniserende woord voor het paar $\{1, k+1\}$ gelijk is aan $n^3 - \frac{3}{2}n^2 + \frac{1}{2}$ voor n is oneven, en $n^3 + \frac{5}{6}n^2 - \frac{4}{3}n + 2$ als n even. Omdat we sowieso dit paar moeten synchroniseren, verwachten we dat:

$$\begin{aligned}
n^3 - \frac{3}{2}n^2 + \frac{1}{2} &\leq r_n && \text{als } n = 2k+1, k \in \mathbb{N} \\
n^3 + \frac{5}{6}n^2 - \frac{4}{3}n + 2 &\leq r_n && \text{als } n = 2k, k \in \mathbb{N}
\end{aligned}$$

En er geldt dat:

$$n^3 + \frac{5}{6}n^2 - \frac{4}{3}n + 2 \geq n^3 - \frac{3}{2}n^2 + \frac{1}{2}$$

Dus $n^3 + \frac{5}{6}n^2 - \frac{4}{3}n + 2 \leq r_n$.

$$r_n \leq 4n^3 \log(n) + \mathcal{O}(n^3)$$

We willen Q synchroniseren met behulp van een synchroniserend woord w , met $|Q| = n$.

Definieer een verzameling $S \subseteq Q$ met $|S| = \alpha$ dan definiëren we $S_w = \bar{\delta}(S, w)$. Dat wil zeggen dat S_w de verzameling toestanden is die we bereiken als we het woord w toepassen op de verzameling S .

Als $|S_w| = 1$ dan is de verzameling S , $|S| > 1$ gesynchroniseerd met behulp van het woord w , want alle toestanden in S worden naar dezelfde toestand gebracht met het woord w .

We gaan gebruik maken van het feit dat we weten hoelang het maximaal duurt om een paar met onderlinge afstand $m - 1$ te synchroniseren, zie lemma 4.16 en stelling 4.15.

Omdat we opzoek zijn naar een bovengrens voor r_n nemen we steeds de slechtst mogelijk S . Dat wil zeggen een $S \subseteq Q$ zodanig dat alle elementen zover mogelijk uit elkaar liggen, want dan duurt het het langst om de verzameling S te synchroniseren. De elementen liggen het verst uit elkaar als ze onderlinge afstand $\frac{n}{\alpha}$ hebben, dit omdat op deze manier alle onderlinge afstanden gelijk zijn. Ze hebben zo de groots mogelijk onderlinge afstand, want stel de afstand tussen $u, v \in S$ maken we eentje groter dan $\exists w \in S$ zodanig dat de onderlinge afstand van u, w of v, w kleiner is geworden. Omdat we alleen maar gehele afstanden kunnen hebben moeten we het naar boven of beneden afronden. Er is altijd een paar dat het snelst synchroniseert namelijk het paar dat het dichtst bij elkaar ligt en er is altijd een of meerdere paar dat het dichtst bij elkaar ligt, namelijk met afstand $\lfloor \frac{n}{\alpha} \rfloor$.

Om heel Q te synchroniseren moeten we elk paar synchroniseren. We definiëren nu het volgende algoritme om een synchroniserend woord te vinden voor Q .

Algoritme:

$S = Q$

$\alpha = n$

$w = \lambda$

While($\alpha > 1$)

 Bepaal p, q met $dist(p, q)$ minimaal.

 Benereer random input tot p, q gesynchroniseerd zijn.

 Bereken S_v

$S = S_v$

$w = wv$

$\alpha = \alpha - 1$

if($\alpha = 1$)

 Print w

Wat dit algoritme eigenlijk doet is dat hij de verzameling $S = Q$ neemt, het paar dat het snelst synchroniseert zoekt en deze synchroniseert met behulp van een woord v . Dan is de verzameling S met 1 afgenomen, waarna hij alle gevonden v 's achter elkaar plakt. Zo gaat het verder tot de verzameling S nog maar uit 1 element bestaat en zo dus de hele verzameling is gesynchroniseerd met behulp van het woord w .

Om de verwachte lengte van het woord w te vinden moeten we de verwachte lengtes van de woorden v vinden.

Met Stelling 4.15 en Lemma 4.16, weten we wat de maximale verwachte lengte van het synchroniserende paar is met onderlinge afstand $m - 1$. We weten dus dat de verwachte lengte van het woord v maximaal is. Voor elke $\alpha = 1, \dots, n$ moet een woord worden gevonden De onderlinge afstand $m - 1$ is afhankelijk van de α , want we hebben gezien dat voor de slechtste S geldt dat de beste onderlinge afstand gelijk is aan $m - 1 = \lfloor \frac{n}{\alpha} \rfloor$ en dus $m = \lceil \frac{n}{\alpha} \rceil + 1 < \frac{n}{\alpha - 1}$.

Dus de verwachte lengte van het woord w is gelijk aan:

Als $n = 2k + 1$

$$\sum_{\alpha=1}^n 4\frac{n^3}{\alpha} - 4(\frac{n^2}{\alpha^2} - \frac{n^2}{\alpha}) + 2(\frac{n^2}{\alpha} - 2\frac{n}{\alpha}) + 2$$

$$< 4n^3 \log(n) + 4n^2 \log(n) - 4n \log(n) + 2n$$

Als $n = 2k$

$$\sum_{\alpha=1}^n 4\frac{n^3}{\alpha} - 4(\frac{n^3}{\alpha^2} - 2\frac{n^2}{\alpha} - n) + 2(\frac{n^2}{\alpha^2} - 4\frac{n}{\alpha}) - 6$$

$$< 4n^3 \log(n) + 4n^2 \log(n) + 4n^2 - 4n \log(n) - 6n$$

dus voor $n \in \mathbb{Z}$ geldt dat:

$$r_n \leq 4n^3 \log(n) + \mathcal{O}(n^3)$$

□

4.4 Synchronisatie van andere automaten

In deze deel-sectie zijn we geïnteresseerd in de verwachte lengte van synchroniserende woorden van algemene synchroniserende automaten.

We werken in deze deel-sectie toe naar een stelling die de maximale verwachte lengte van een woord tot synchronisatie van boven en onder begrensd.

Hiervoor hebben we de volgende definitie nodig:

Definitie 4.18. Zij $r(\mathcal{A}) = \mathbb{E}(\text{tijd tot synchronisatie van } \mathcal{A})$.

Dan definiëren we $R(n)$ als $R(n) = \max\{r(\mathcal{A}) \mid \mathcal{A} \text{ synchroniserend}\}$

Stel we nemen een automaat $\mathcal{A}$ waarvan we weten dat als we kijken naar het deterministische geval het woord w synchroniserend is, dan weten we dat dit woord w ook in het geval van random input zou gaan werken. We zijn nu geïnteresseerd in de verwachte aantal letters die we nodig hebben tot we het woord w hebben toegepast. Want als we woord w hebben toegepast dan is de automaat gesynchroniseerd.

De stelling hieronder verteld ons iets over de verwachte wachttijd tot het woord w verschijnt.

Stelling 4.19. Zij $w \in \{a, b\}^*$ en zij $x_1, x_2, \dots$ random onafhankelijke letters met kans $\mathbb{P}(x_i = a) = \mathbb{P}(x_i = b) = \frac{1}{2}$.

Zij $T_w = \min\{k \mid w \text{ komt voor in } x_1 x_2 \dots x_k\}$ dan geldt dat $\mathbb{E}(T_w) = 2^{|w|} + \mathbb{E}(T_u)$.

Met $u \neq w$ het langste sub-woord dat zowel prefix als suffix van w is.

Voorbeeld 4.20. We willen weten wat $\mathbb{E}(T_w)$ is met $w = abb$, dan is het λ zowel de prefix als de suffix van het woord w , dus $u = \lambda$ en er geldt dus dat:

$$\mathbb{E}(T_w) = 2^{|w|} + \mathbb{E}(T_u) = 2^3 + 0 = 8$$

Voorbeeld 4.21. We willen nu weten wat $\mathbb{E}(T_w)$ met $w = abba$ het langst mogelijk subwoord dat zowel de prefix en suffix van w is gelijk aan $u = a$. Dus er geldt:

$$\mathbb{E}(T_w) = 2^{|w|} + \mathbb{E}(T_u) = 2^4 + 2^1 + \mathbb{E}(T_\lambda) = 10$$

Voorbeeld 4.22. We willen nu weten wat $\mathbb{E}(T_w)$ met $w = bbbb$ het langste mogelijk subwoord dat zowel de prefix als suffix is van w is gelijk aan $u = bbb$.

$$\begin{aligned} \mathbb{E}(T_w) &= 2^4 + \mathbb{E}(T_{bbb}) \\ &= 2^4 + 2^3 + \mathbb{E}(T_{bb}) \\ &= 2^4 + 2^3 + 2^2 + \mathbb{E}(T_b) \\ &= 2^4 + 2^3 + 2^2 + 2^1 + \mathbb{E}(T_\lambda) \\ &= 2^4 + 2^3 + 2^2 + 2^1 + 0 \\ &= 16 + 8 + 4 + 2 \\ &= 30 \end{aligned}$$

Met behulp van stelling 4.19 kunnen we het volgende zeggen over over de verwachte aantal letters die we nodig hebben tot we woord w tegen komen.

Lemma 4.23. $\mathbb{E}(\text{tijd tot synchronisatie}) \leq 2^{\frac{n^3-n}{6}+1} - 2$

Opmerking: Als w een kortst synchroniserend woord is dan geldt er dat $\mathbb{E}(T_w) \geq \mathbb{E}(\text{tijd tot synchronisatie})$.

Bewijs. Uit stelling 4.19, zien we wat de verwachting is tot we w tegenkomen. In het slechtste geval wachten we op een woord met alleen maar de zelfde letter, zoals we zagen in voorbeeld 4.22.

Als dat het geval is dan geldt dat: $\mathbb{E}(T_w) = \sum_{k=1}^{|w|} 2^k = 2^{|w|+1} - 2$. In stelling 3.4 zagen we dat $C(n) \leq \frac{n^3-n}{6}$ en dus geldt $|w| \leq \frac{n^3-n}{6}$ als we dit invullen krijgen we:

$$\mathbb{E}(\text{tijd tot synchronisatie}) \leq 2^{\frac{n^3-n}{6}+1} - 2.$$

Hiermee is het gevraagde bewezen.

□

Opmerking: Er geldt niet dat $\mathbb{E}(T_w) \geq 2^{|w|}$, want stel we hebben twee synchroniserende woorden w_1 en w_2 zodanig dat $|w_1| \leq |w_2|$ dan geldt er dat $\mathbb{E}(T_{w_1}) \geq \mathbb{E}(T_{w_1} \circ f T_{w_2})$.

We kunnen de bovengrens die we gevonden hebben in lemma 4.23 nog scherper maken. Er geldt namelijk het volgende.

Stelling 4.24. $\mathbb{E}(\text{tijd tot synchronisatie}) \leq (n-1)(2^{\binom{n}{2}+1} - 2) = 2^{\frac{1}{2}n^2 + o(n)}$

Bewijs. [6] Zoals we ook in het bewijs voor lemma 3.3 zagen geldt het volgende: $\forall q_1, q_2 \in Q \exists w$ met $|w| \leq \binom{n}{2}$ zodanig dat $\delta(q_1, w) = \delta(q_2, w)$. Neem nu $S \subseteq Q, |S| = k$. Ook definiëren we weer $S_w = \{r \in Q \mid \delta(r', w) = r, r' \in S\}$. We weten dat er voor elk paar dat we vinden in S er een woord w bestaat zodanig dat $|w| \leq \binom{n}{2}$. We weten ook dat voor deze w dan het volgende geldt: $|S_w| < |S|$, want er is minstens een paar dat synchroniseert dus we weten ook dat $|S_w| \leq |S| - 1$. Dit gaan we gebruiken om de lengte van een synchroniserend woord af te schatten.

We beginnen met $S = Q$. Dan weten we dat we een synchroniserend woord w kunnen vinden voor een paar in S zodanig dat $|w| \leq \binom{n}{2}$. Daarna bekijken we S_w waarvan we weten dat deze verzameling minstens een kleiner is dan S . Vervolgens nemen we $S = S_w$, dan kunnen we weer een woord w vinden met $|w| \leq \binom{n}{2}$ zodat een paar in de verzameling S wordt gesynchroniseerd. Vervolgens nemen we weer $S = S_w$. Dit principe herhalen we totdat $|S| = 1$. We willen een synchroniserend woord v voor de verzameling Q vinden. Als we nu v nemen als het woord dat alle gevonden w 's achter elkaar toepast, dan hebben we een synchroniserend woord voor de verzameling Q . Maar hoelang is het woord v dan? Voor elke gevonden woord w om een paar te synchroniseren gold dat $|w| \leq \binom{n}{2}$. We moeten $n-1$ keer een paar synchroniseren om bij $|S| = 1$ uit te komen. Volgens stelling 4.19 verwachten $2^{|w|+1} - 2$ stappen tot we woord w gezien hebben (zie bewijs Lemma 4.23 voor de uitleg). We moeten nu $n-1$ keer wachten op een woord van lengte $|w| \leq \binom{n}{2}$, dus geldt er dat: $\mathbb{E}(\text{tijd tot synchronisatie}) \leq (n-1)(2^{\binom{n}{2}+1} - 2)$. $\square$

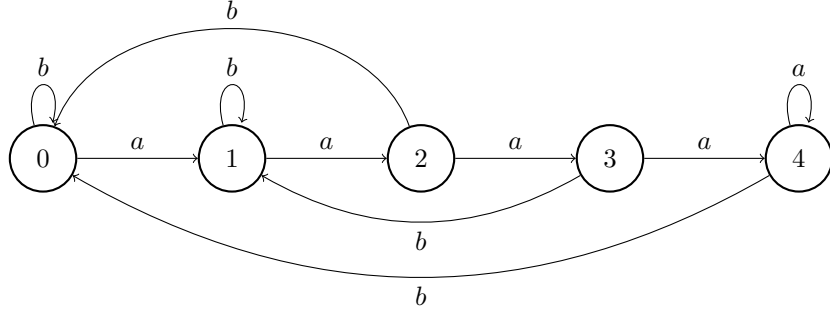
Nu we een bovengrens hebben gevonden voor de maximale verwachte lengte tot we het synchroniserende woord tegen komen, zijn we natuurlijk ook geïnteresseerd in een ondergrens. Maar zoals je ziet in de opmerking is dat nog niet zo gemakkelijk, omdat er altijd nog een synchroniserend woord te vinden is voor de synchroniserende automaat en die misschien eerder verschijnt. Om een ondergrens te vinden moeten we een automaat zien te vinden, met een synchroniserend woord dat er heel lang over doet om met random input te krijgen. Zoals we eerder hebben gezien, in voorbeeld 4.22, is dit een woord die alleen maar de zelfde letter heeft. We gaan dus opzoek naar een automaat die er zo lang mogelijk over doet om te synchroniseren, met een woord w die er lang over doet om te verkrijgen in het geval van random input.

Voorbeeld 4.25. De transitie functie voor de automaat met een synchroniserende woord die een ondergrens vormt voor het maximum van $\mathbb{E}(\text{tijd tot synchronisatie})$ is:

$$\begin{aligned} \delta : Q \times \Sigma &\longrightarrow Q : \\ \delta(x, a) &= x + 1 \quad \text{voor } 0 \leq x \leq n-2 \\ \delta(n-1, a) &= n-1 \\ \delta(x, b) &= 0 \quad \text{als } x = 2k, k \in \mathcal{N} \\ \delta(x, b) &= 1 \quad \text{als } x = 2k+1, k \in \mathcal{N} \end{aligned} \tag{21}$$

Het synchroniserende woord dat hoort bij deze automaat is: $w = a^{n-1}$.

In figuur 11 staat een voorbeeld voor deze automaat met $n = 5$. Zoals we zagen in het bewijs van Lemma 4.23 is de verwachte tijd tot we woord w tegenkomen is gelijk aan: $\sum_{k=1}^{|w|} 2^k = 2^{|w|+1} - 2$. Als we nu de lengte van het gevonden woord w invullen krijgen we: $2^n - 2$.



Figuur 11: Slechte automaat

Dit voorbeeld leidt tot een ondergrens voor $\mathbb{E}(\text{tijd tot synchronisatie})$. We hebben hier namelijk een automaat die er lang over doet om te synchroniseren en een woord w waarop we lang moeten wachten tot we verwachten hem zien. Dus er geldt dat:

$$2^n - 2 \leq \mathbb{E}(\text{tijd tot synchronisatie}) \quad (22)$$

Alles samen genomen kunnen we nu de volgende stelling opstellen over $R(n)$ voor alle automaten.

Stelling 4.26. $2^n - 2 \leq R(n) \leq 2^{\frac{1}{2}n^2 + o(n)}$

Bewijs. Het bewijs volgt uit stelling 4.24 en voorbeeld 4.25. Want stelling 4.24 zegt dat $\mathbb{E}(\text{tijd tot synchronisatie}) \leq 2^{\frac{1}{2}n^2 + o(n)}$, dan moet ook gelden dat het maximum kleiner gelijk zijn dan $2^{\frac{1}{2}n^2 + o(n)}$. En met voorbeeld 4.25 hebben we een automaat gevonden waarvoor geldt dat $2^n - 2 \leq \mathbb{E}(\text{tijd tot synchronisatie})$ en dan moet het maximum ook groter of gelijk aan $2^n - 2$. $\square$

5 Resultaten

In de afgelopen secties hebben we het gehad over twee verschillende gevallen. Als eerste het deterministische geval, zie sectie 2 en 3, en als tweede over het geval van random input, zie sectie 4. We kunnen met behulp van deze twee gevallen interessante conclusies trekken.

5.1 Černý automaat

Laten we beginnen met de Černý automaat $\mathcal{C}_n$, zoals we in sectie 3.2 zagen voldoet de Černý automaat aan het Černý vermoeden. In het deterministische geval is de lengte van het kortste synchroniserende woord w voor de automaat met n toestanden gelijk aan $|w| = (n - 1)^2$.

Als we kijken naar het geval van random input, dan geldt dat:

$$r_n \leq 4n^3 \log(n) + \mathcal{O}(n^3)$$

5.2 Slechte automaat

We kijken in deze deel-sectie naar de automaat gegeven door de transitie functie die gegeven is in (21).

Als eerste kijken we naar het deterministische geval. We zien hier uit voorbeeld 4.25 dat de lengte van het kortste synchroniserende woord w voor de automaat met n toestanden gelijk is aan $|w| = n - 1$.

Nu kijken we naar het geval van random input. Dan zien we in voorbeeld 4.25 dat

$$\mathbb{E}(\text{tijd tot synchronisatie}) = 2^n - 2.$$

5.3 Andere automaten

In deze deel-sectie kijken we niet specifiek naar een bepaald soort automaat, maar naar alle andere automaten in het algemeen. We kijken weer eerst naar het deterministische geval. In sectie 3.1 hebben we gezien dat $(n - 1)^2 \leq C(n) \leq \frac{n^3 - 1}{6}$.

Als we nu kijken naar het geval van random input dan geldt er dat:

$$2^n - 2 \leq R(n) \leq 2^{\frac{1}{2}n^2 + o(n)}.$$

5.4 Conclusie

Wat opvalt is dat de lengte van het synchroniserende woord in het deterministische weinig zegt de verwachting tot we de automaat hebben gesynchroniseerd in het geval van random input. Kijk maar naar de Černý automaat en de automaat die hoort bij de transitie functie gegeven in (21). De lengte van het synchroniserende woord is bij de Černý automaat kwadratisch groter, maar de verwachting tot we de automaat synchroniseren in het geval van random input is bij de Černý automaat weer veel kleiner dan dat van de slechte automaat.

Referenties

- [1] Jurriaan Rot. *Talen en Automaten*. URL: <http://cs.ru.nl/~jrot/TnA2017/>.
- [2] M.V. Volkov. „Synchronizing Automata and the Cerny Conjecture”. In: *Springer-Verlag Berling Heidelberg* (2008), p. 11–27. DOI: <http://www.math.uni.wroc.pl/~kisiel/auto/volkov-surv.pdf>.
- [3] P.Frankl. „An Extremal Problem for two Families Of Sets”. In: *Europ. J. Combinatorics* 3.3 (1982), p. 125–127. DOI: <https://www.sciencedirect.com/science/article/pii/S0195669882800255>.
- [4] David Eppstein. „Reset Sequences for Monotonic Automata”. In: (1990), p. 5. DOI: <https://www.ics.uci.edu/~eppstein/pubs/Epp-SJC-90.pdf>.
- [5] J.R. Norris. *Markov Chains*. Cambrigde Series in Statistical and Probabilistic Mathematics. 1997, p. 12–18.
- [6] Dhr. Dr. Ir. H. Don.