

WISKUNDIGE ASPECTEN VAN DE SUDOKU

Maartje Geurts

3002268

Wiskunde, Bachelor

Begeleider: Wieb Bosma

30 juni 2013

Radboud Universiteit Nijmegen



Samenvatting

In deze scriptie worden verschillende wiskundige eigenschappen van sudoku's en van sudokupuzzels bekeken. Verder wordt er een algoritme gegeven dat een manier beschrijft om een sudokupuzzel met pen en papier op te lossen. Op basis van dit algoritme, wordt later een manier beschreven om de moeilijkheidsgraad van een sudokupuzzel te bepalen.

Inhoudsopgave

1	Inleiding	4
2	Sudoku	5
2.1	Sudoku's tellen	7
2.1.1	Sudoku van orde 2	8
2.1.2	Sudoku van orde 3	11
2.2	Sudoku's tellen via de grafentheorie	13
3	Structuur van de sudokupuzzel	18
4	Oplossingsstrategiën	21
5	De moeilijkheidsgraad bepalen	28

1 Inleiding

Eind jaren '70 werd de sudokupuzzel (die toen nog Number Place heette) voor het eerst gepubliceerd. Howard Garns was de bedenker. Toen de puzzel in 1984, 5 jaar na de eerste publicatie, door Nikoli in Japan werd geïntroduceerd begon de puzzel langzaam meer populariteit te verwerven. Nikoli heeft de puzzel toen de naam Sudoku gegeven. Sudoku is een samenvoeging van “suuji wa dokushin ni kagiru” wat vertaald kan worden als “de getallen moeten alleen zijn”. Su betekent getal en Doku betekent alleen. In de zomer van 2005 werd de puzzel voor het eerst in Nederland gepubliceerd en sindsdien kun je geen krant meer openslaan zonder een sudokupuzzel te zien. In eerste instantie werd vooral de 9-bij-9-variant gepubliceerd, maar al snel kwamen er heel veel verschillende variaties van de sudoku.

Voor mijn bachelorscriptie heb ik me verdiept in de wiskundige aspecten van de sudoku. Dit leek me leuk omdat ik me voor de verandering eens bezig wilde houden met een wiskundig onderwerp dat ook bekend is onder de niet-wiskundigen. Voor het merendeel van de vakken binnen de wiskunde bachelor geldt dat ik er alleen over kan praten met mensen binnen de opleiding. Het leuke aan mijn scriptie vind ik, dat ik het (tot op zekere hoogte) uit kan leggen aan iedereen die wel eens een sudokupuzzel heeft gemaakt.

De grote bekendheid van het onderwerp, bracht hier en daar ook juist moeilijkheden met zich mee. Er zijn heel veel wetenschappelijke artikelen geschreven over sudoku's, maar ik ben nog geen enkel artikel tegen gekomen waarin een exacte definitie van de sudoku of de sudokupuzzel staat. In het eerste hoofdstuk zullen deze begrippen dan ook direct gedefinieerd worden. Verder worden er in dat hoofdstuk enkele manieren bekeken om het aantal sudoku's te bepalen. Aan het eind van dit hoofdstuk is de structuur van de sudoku uitgebreid behandeld. De rest van de hoofdstukken zijn voornamelijk gericht op de structuren die zijn te vinden in de sudokupuzzels. Zo wordt in hoofdstuk 3 behandeld hoe zogenaamde unavoidable sets samenhangen met de uniciteit van sudokupuzzels. In hoofdstuk 4 worden o.a. pre-emptive sets geïntroduceerd om een wiskundige benadering te geven voor de menselijke manier om een sudokupuzzel op te lossen. Aan het eind van hoofdstuk 4 wordt een oplossingsalgoritme gegeven dat in hoofdstuk 5 gebruikt zal worden om de moeilijkheidsgraad van een sudokupuzzel te bepalen.

2 Sudoku

Om maar meteen met de deur in huis te vallen, beginnen we direct met de definitie van een geheel gegeneraliseerde sudoku.

Definitie 2.1. Een *gegeneraliseerde sudoku over symbolenverzameling S* is een verzameling $\{X, \{Y_i\}_{i \in I}, \varphi\}$ waarbij X een niet-lege verzameling cellen is met een surjectieve afbeelding $\varphi : X \rightarrow S$, zodat $\{Y_i\}_{i \in I}$ een familie deelverzamelingen van X is met:

- $\bigcup_{i \in I} Y_i = X$
- $\forall i \in I$ is $\varphi|_{Y_i} : Y_i \rightarrow S$ bijectief.

Opmerking 2.2. Een Latijns vierkant is een sudoku waarbij $\{Y_i\}_{i \in I}$ precies de rijen en kolommen van het vierkant zijn.

In bovenstaande definitie van de sudoku is φ al bekend. Voor elke cel uit X is dus al gegeven welk symbool uit S in die cel moet komen te staan. De sudoku is dus de oplossing van een sudokupuzzel zoals we die bijvoorbeeld in de krant tegen komen. Hieronder staat de definitie van een sudokupuzzel gegeven:

Definitie 2.3. Een *sudokupuzzel over S* is een verzameling $\{X, Z, \{Y_i\}_{i \in I}, \varphi_0\}$ met $Z \subseteq X$ waarvoor geldt:

- $\varphi_0 : Z \rightarrow S$
- $\{(z, \varphi_0(z)) \mid z \in Z\}$ is de verzameling *clues*
- $\exists! \varphi$ zodat:
 - $\varphi|_Z = \varphi_0$
 - $\{X, \{Y_i\}_{i \in I}, \varphi\}$ is een sudoku

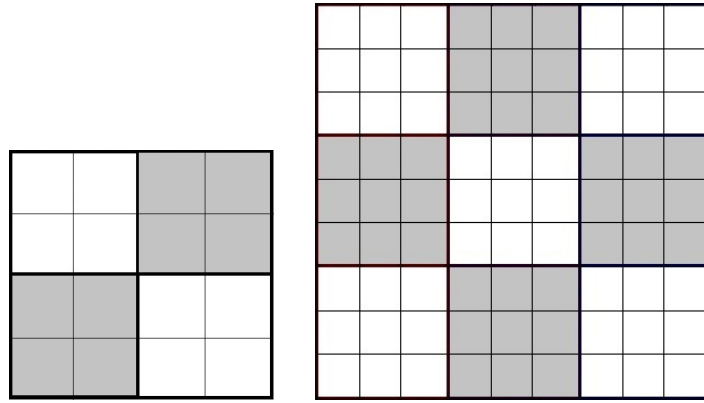
Definitie 2.4. Zij $\{X, Z, \{Y_i\}_{i \in I}, \varphi_0\}$ een sudokupuzzel. Het *oplossen van een sudokupuzzel* is het definiëren van een familie $\{\varphi_i\}_{i \in \{0, 1, \dots, \#X - \#Z\}}$ van partiële functies met

- φ_0 als in de gegeven puzzel
- $\varphi_{\#X - \#Z} = \varphi$ zodat $\{X, \{Y_i\}_{i \in I}, \varphi\}$ een sudoku is
- $i \leq j \Rightarrow \text{dom}(\varphi_i) \subseteq \text{dom}(\varphi_j)$
- $\forall \{i \in 0, 1, \dots, \#X - \#Z\}$: als $i \leq j$ en $x \in \text{dom}(\varphi_i) \Rightarrow \varphi_i(x) = \varphi_j(x)$

Opmerking 2.5. Voor alle k in $\{i \in 0, 1, \dots, \#X - \#Z\}$ geldt nu dat $\text{dom}(\varphi_k) \subseteq X$ precies alle cellen uit X zijn die al ingevuld zijn (dat zijn dus alle clues en de handmatig ingevulde symbolen) en dat $\text{dom}(\varphi_k) \setminus \text{dom}(\varphi_{k-1})$ is precies de cel die als k^{de} is ingevuld.

De definitie van de gegeneraliseerde sudoku bovenaan dit hoofdstuk dekt alle verschillende soorten sudoku's. De meest voorkomende sudoku is echter een specifiek geval, namelijk die waarbij X een $n^2 \times n^2$ -matrix over S is die is onderverdeeld in n^2 niet overlappende blokken van $n \times n$. Deze blokken zou je ook kunnen zien als disjuncte $n \times n$ -deelmatrices.

In onderstaand figuur staan de blokken weergegeven voor de gevallen $n = 2$ en $n = 3$.



Figuur 1: $n = 2 \Rightarrow$ 4 blokken
Figuur 2: $n = 3 \Rightarrow$ 9 blokken

Definitie 2.6. Een gegeneraliseerde sudoku heet een *sudoku van orde n* als $S = \{1, \dots, n^2\}$, $X \in \text{Mat}_{n^2 \times n^2}(S)$ en $I = \{1, \dots, 3n^2\}$ zodat $\{Y_i\}_{i \in I}$ precies de rijen, kolommen en blokken van de matrix zijn.

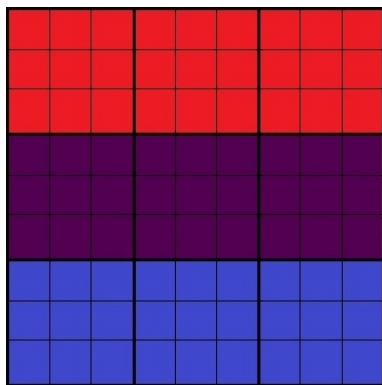
Vanaf nu zal, tenzij anders vermeld staat, met ‘sudoku’ verwezen worden naar een sudoku van orde 3 zoals in bovenstaande definitie beschreven staat. In dat geval geldt dus dat $S = \{1, \dots, 9\}$, $X \in \text{Mat}_{9 \times 9}(S)$ waarbij $I = \{1, \dots, 27\}$ zodat $\{Y_i\}_{i \in I}$ precies alle 9 rijen, 9 kolommen en 9 blokken van de matrix bevat.

2.1 Sudoku's tellen

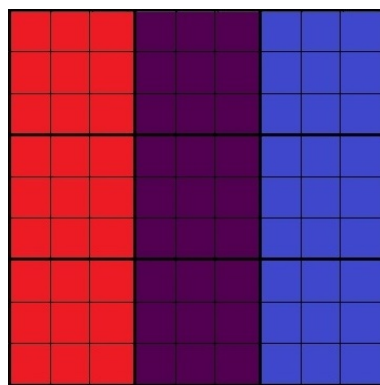
Er zijn ontzettend veel sudoku's. Hoe groter het aantal cellen van een sudoku is, hoe moeilijker het is om het aantal sudoku's te bepalen. In deze paragraaf staat beschreven hoe de sudoku's van orde 2 en die van orde 3 bepaald kunnen worden. Maar voordat we zover zijn, eerst nog wat anders.

Tussen alle verschillende sudoku's zitten heel veel sudoku's die “erg veel op elkaar lijken”. Want als je een willekeurige sudoku bekijkt en je verwisselt overal het symbool s_1 met het symbool $s_2 \neq s_1$ en andersom, dan heb je nog steeds een sudoku en deze sudoku is ongeveer hetzelfde als de oorspronkelijke sudoku. Er zijn nog een aantal andere manieren waarop je een sudoku kunt aanpassen zonder dat de sudoku essentieel anders wordt. Om hier verder op in te gaan moet eerst het volgende begrip geïntroduceerd worden:

Definitie 2.7. Een *blokrij* is een verzameling van drie blokken die recht naast elkaar staan. Een *blokkolom* is een verzameling van drie blokken die recht onder elkaar staan.



Figuur 3: Blokrijen in de sudoku van orde 3



Figuur 4: Blokkolommen in de sudoku van orde 3

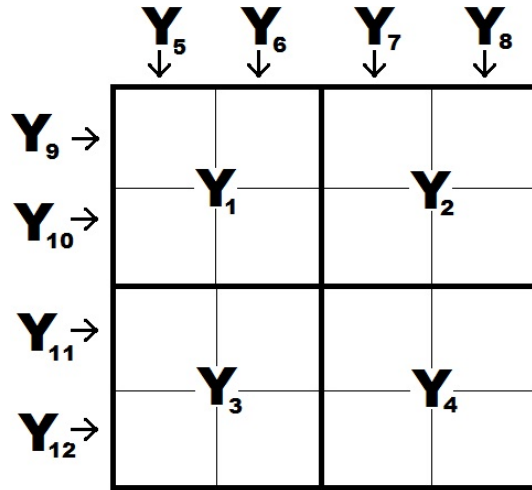
Het is nu gemakkelijk in te zien dat je twee rijen binnen een blokrij met elkaar kunt verwisselen zonder dat de sudoku daardoor essentieel verandert. Vanaf nu noemen we twee sudoku's *essentieel verschillend* of *niet-equivalent* als ze *niet* aan elkaar gelijk kunnen worden door middel van symboolverwisselingen, rij- of kolomtranspositie binnen een blokrij of blokkolom, blokrij- of blokkolomtranspositie (verwisselingen van de blokrijen of blokkolommen) of transponeren (spiegeling in de diagonaal van de matrix).

In deze paragraaf bekijken we verschillende manieren om sudoku's te tellen.

2.1.1 Sudoku van orde 2

Om wat meer inzicht te krijgen in het tellen van sudoku's, bekijken we eerst de sudoku van orde 2. Die gaan we tellen door simpelweg te bekijken op hoeveel manieren je de matrix in kan vullen zodat er een sudoku ontstaat.

De sudoku van orde 2 over S is formeel te beschrijven als $\{X, \{Y_i\}_{i \in I}, \varphi\}$ waarbij $S = \{1, \dots, 4\}$, $X \in \text{Mat}_{4 \times 4}(S)$ en $I = \{1, \dots, 12\}$ zodat $\{Y_i\}_{i \in I}$ precies de rijen, kolommen en blokken van de matrix zijn. In de afbeelding hieronder zijn de elementen uit $\{Y_i\}_{i \in I}$ aangegeven.



Figuur 5: $\{Y_i\}_{i \in I}$ van de sudoku van orde 2

Omdat we de symbolen onderling zouden kunnen verwisselen zijn er $4! = 24$ equivalente mogelijkheden om Y_1 in te vullen.

Zonder verlies van algemeenheid kunnen we uit gaan van de volgende manier:

1	2
3	4

Figuur 6: Y_1

Y_5 kan nu, net als Y_9 , ingevuld worden op 2 verschillende manieren. Omdat we de laatste twee kolommen en de laatste twee rijen altijd onderling mogen

verwisselen, kunnen we nu zonder verlies van algemeenheid aannemen dat de laatste twee cellen van de eerste rij en die van de eerste kolom er zo uit zien als in figuur 7. Er zijn dus 2×2 equivalente mogelijkheden om deze aangegeven cellen in te vullen.

1	2	3	4
3	4		
2			
4			

Figuur 7: Y_5 en Y_9 ingevuld

1	2	3	4
3	4		
2		4	
4			

Figuur 8: 4 invullen

Het is nu gemakkelijk in te zien dat er maar één mogelijke cel is in Y_4 waar het symbool 4 ingevuld kan worden. (figuur 8) Ook zie je na even nadenken dat je een soortgelijke stap had kunnen maken als Y_5 en Y_9 anders gekozen zouden zijn.

Om de sudoku af te maken, kunnen we nu voor elke nog lege cel nagaan welke sudoku of sudoku's we krijgen als we er een bepaald symbool invullen. Als we een symbool kiezen voor de cel helemaal rechts onderin, dan ligt de sudoku verder vast en hoeven we dus de minste mogelijkheden af te gaan. Voor deze cel zijn drie mogelijkheden, namelijk 1, 2 of 3. (Als we met een andere cel zouden beginnen zouden er meer stappen ondernomen moeten worden, maar uiteindelijk komen alleen de onderstaande sudoku's als oplossing tevoorschijn.) Met deze stap zien we dus dat er nog **3** keuzemogelijkheden zijn.

1	2	3	4
3	4		
2		4	
4			1

1	2	3	4
3	4		
2		4	
4			2

1	2	3	4
3	4		
2		4	
4			3

Figuur 9: De 3 mogelijkheden

De sudoku's in figuur 9 zijn alle drie op een eenduidige manier te completeren. In figuur 10 staat het resultaat.

1	2	3	4
3	4	1	2
2	1	4	3
4	3	2	1

1	2	3	4
3	4	2	1
2	1	4	3
4	3	1	2

1	2	3	4
3	4	1	2
2	3	4	1
4	1	2	3

Figuur 10: De ingevulde roosters

Bovenstaande sudoku's zijn alle drie verschillend, maar als we bij de middelste sudoku de symbolen 2 en 3 verwisselen en vervolgens de sudoku transponeren, zien we dat deze gelijk is aan de meest rechtse sudoku.

1	2	3	4
3	4	1	2
2	1	4	3
4	3	2	1

1	2	3	4
3	4	2	1
2	1	4	3
4	3	1	2

Figuur 11: Twee inequivalente sudoku's

De twee overgebleven sudoku's verschillen alleen in de aangegeven rode cellen. De symbolen in deze cellen kunnen met de geldige operaties niet verwisseld worden zodat de rest van de sudoku geheel hetzelfde blijft. Deze twee sudoku's zijn dus inequivalent.

Opmerking 2.8. In het algemeen is het nog niet zo eenvoudig om aan te tonen dat twee sudoku's *niet* equivalent zijn.

We zien nu dus dat er in totaal maar 2 essentieel verschillende sudoku's van orde 2 zijn. Hoewel we uit $4!$ mogelijkheden konden kiezen voor Y_1 , vervolgens op 2 manieren de eerste kolom af konden maken en op 2 manieren de eerste rij en daarna waren er nog 3 manieren om de sudoku te completeren. In totaal zijn er dus $4! \times 2 \times 2 \times 3 = 288$ verschillende manieren zijn om een zo'n sudoku in te vullen.

2.1.2 Sudoku van orde 3

Voor de sudoku van orde 3 is het een stuk moeilijker om het aantal mogelijke sudoku's te tellen.

In juni 2005 kwamen Bertram Felgenhauer en Frazer Jarvis met een artikel waarin zij beschreven hoe het aantal sudoku's bepaald kan worden. Met het nodige denkwerk en na het ontwikkelen van een slimme brute-force methode, zijn ze erachter gekomen dat er precies $6670903752021072936960 \approx 6,67 \cdot 10^{21}$ sudoku's zijn.

Gegeven dit aantal, kunnen we nu na gaan hoeveel daarvan essentieel verschillend zijn. Hierover gaat de volgende propositie.

Propositie 2.9. *Er zijn $5472730538 (\approx 5,47 \cdot 10^9)$ essentieel verschillende sudoku's van orde 3.*

Bewijs. Bekijk per operatie op hoeveel verschillende manieren je deze toe kunt passen op de sudoku.

- Blokrijverwisselingen: Twee blokrijen kunnen met elkaar verwisseld worden. In totaal zijn er $3! = 6$ manieren kunnen verwisselen.
- Blokkolomverwisselingen: Op dezelfde manier als bij de blokrijen kunnen we de blokkolommen ook op $3! = 6$ manieren verwisselen.
- Rijverwisselingen: Twee rijen binnen een blokrij kunnen $3! = 6$ manier verwisseld worden. Aangezien er 3 blokrijen zijn, zijn er in totaal 6^3 rijverwisselingen mogelijk.
- Kolomverwisselingen: Op dezelfde manier als bij de rijverwisselingen, zijn er ook 6^3 kolomverwisselingen mogelijk.
- Transponeren: Omdat de sudoku ook getransponeerd kan worden zonder dat er een essentiële verandering plaats vindt, geeft deze operatie een factor 2.
- Symboolverwisselingen: Alle symbolen van een soort kunnen verwisseld worden met alle symbolen van een ander soort. Omdat er 9 verschillende symbolen zijn, zijn er $9! = 362880$ manieren om deze symbolen onderling te verwisselen.

Combinaties van bovenstaande operaties geeft $9! \times 6 \times 6 \times 6^3 \times 6^3 \times 2$ mogelijkheden om een sudoku te veranderen zonder dat deze essentieel verschillend wordt.

Dit betekent echter niet dat de symmetriegroep G van de sudoku ook zoveel elementen bevat. Ten eerste verschilt de operatie die de symbolen verwisselt van de andere operaties, omdat de andere operaties paren (x, s) verwisselt (de ingevulde cellen) en de symboolverwisselingen werken slechts op de symbolen. De symmetriegroep G bevat alleen de operaties die werken op de ingevulde cellen. Ten tweede worden er in bovenstaande operaties een aantal groepselementen dubbel geteld. We zien namelijk dat het verwisselen van kolom 1 en kolom 2 gelijk is aan het eerst transponeren van de sudoku, vervolgens het verwisselen van rij 1 en rij 2 en vervolgens de sudoku weer transponeren. Zo is elke kolomtranspositie te schrijven in termen van transponaties en rijtransposities en elke blokkolomtranspositie is te schrijven in termen van blokrijtransposities en transponaties.

G bevat dus de rijtransposities, blokrijtransposities, transponatie en alle combinaties daarvan. De orde van G is dus $6^3 \times 6 \times 2 = 2592$.

We bekijken nu X de verzameling van alle sudoku's van orde 3. $X \backslash G$ is dan het aantal banen van X onder werking van symmetriegroep G , wat dus precies het aantal essentieel verschillende elementen van X is. Met het lemma van Burnside is dit aantal $X \backslash G$ te bepalen en wel op de volgende manier:

$$\#X \backslash G = \frac{1}{\#G} \sum_{g \in G} \#X^g$$

Hierbij bevat X^g de elementen in X die gedekt zijn door g . Oftewel de elementen $x \in X$ zodat gx (eventueel na symboolverwisselingen) gelijk is aan zichzelf. Voor $g = e$, de eenheid, geldt natuurlijk dat $\#X^g = \#X$. Voor de andere elementen uit G is het meer werk om $\#X^g$ te bepalen. Gelukkig hebben *Russell en Jarvis* (2005) $\#X \backslash G$ al een keer bepaald. Ze kwamen tot de conclusie dat er 5472730538 banen en dus 5472730538 essentieel verschillende sudoku's zijn. $\square$

Opmerking 2.10. Alle andere operaties, zoals bijvoorbeeld de sudoku een kwartslag draaien of spiegelen in de verticale as, zijn combinaties van de hierboven genoemde operaties en zijn dus ook bevat in de symmetriegroep.

2.2 Sudoku's tellen via de grafentheorie

In de vorige paragraaf staat beschreven hoe het aantal sudoku's bepaald kan worden. Net als met zoveel vraagstukken in de wiskunde, kan ook dit vraagstuk op meerdere manieren opgelost worden. De methode die in deze paragraaf behandeld wordt, is die via de grafentheorie.

Om deze methode te kunnen bekijken, zal er allereerst een link gelegd moeten worden tussen een sudoku en een graaf. Dat kan op de volgende manier:

Definitie 2.11. Zij $\{X, \{Y_i\}_{i \in I}, \varphi\}$ een sudoku. De bijbehorende *sudokugraaf* is een graaf $G=(V, E)$ waarbij:

- $V = X$ is de verzameling punten
- $E = V \times V$ is de kantenverzameling met $(x, y) \in E \Leftrightarrow \exists Y_i \text{ zodat } \{x, y\} \subseteq Y_i$

De sudokugraaf van een sudoku van orde 2 is nu een 7-reguliere graaf op 16 punten. Elk punt (elke cel) is namelijk verbonden met de punten die horen bij de cellen uit dezelfde rij, dezelfde kolom en hetzelfde blok.

De sudokugraaf van een sudoku van orde 3 is een 20-reguliere graaf op 81 punten.

Het tellen van het aantal sudoku's kunnen we nu zien als een kleuringsprobleem in de grafentheorie.

Definitie 2.12. Een *kleuring van een graaf met $\#S$ kleuren* is een afbeelding $\theta : X \rightarrow S$.

Als de graaf gekleurd wordt met precies $\#S$ kleuren, dan moet θ dus surjectief zijn.

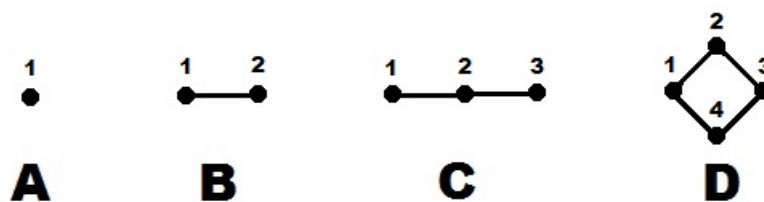
Definitie 2.13. Een kleuring van een graaf heet *correct* of *juist* $\Leftrightarrow \forall x, y \in X$ met $(x, y) \in E$ geldt dat $\theta(x) \neq \theta(y)$.

Het aantal sudoku's is nu gelijk aan het aantal correcte kleuringen van de sudokugraaf met precies $\#S$ verschillende kleuren.

En het aantal manieren waarom je de graaf kan kleuren met t kleuren, is te bepalen met behulp het chromatische polynoom van de graaf.

Definitie 2.14. Het *chromatisch polynoom* van een graaf is een polynoom $\mathcal{P}_G : \mathbb{N} \rightarrow \mathbb{N}$ dat het aantal juiste kleuringen van een graaf telt als functie van het aantal kleuren.

Voor kleine grafen is het vaak niet zo ingewikkeld om dit polynoom te bepalen. Op de volgende pagina staan 4 kleine grafen met daarbij uitleg over hoe het chromatisch polynoom van die grafen bepaald wordt.



Figuur 12: 4 kleine grafen als voorbeeld

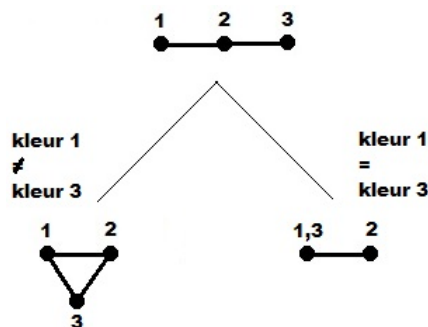
A Het aantal manieren om deze graaf te kleuren met x kleuren, is precies x . Dus $\mathcal{P}_A(x) = x$.

B Voor punt 1 uit deze graaf, kunnen we x kleuren gebruiken. Voor punt 2 mogen we alle kleuren gebruiken, behalve de kleur die we voor punt 1 hebben gebruikt. Dus voor het tweede punt zijn nog $x - 1$ mogelijke kleuren over. Voor deze graaf geldt dus dat $\mathcal{P}_B(x) = x(x - 1) = x^2 - x$.

C Punt nummer 1 kan met x kleuren ingekleurd worden. Voor punt 2 zijn dan nog $x - 1$ kleuren over. Nu zijn we bij punt 3 aangekomen en zien we dat we nu twee keuzes hebben.

(a) Punt 3 krijgt dezelfde kleur als punt 1. Nu zouden we deze twee punten dus net zo goed kunnen samenvoegen, want alle buurpunten van punt 1 en 3 mogen niet diezelfde kleur krijgen als punt 1 en 3 zelf.

(b) Punt 3 krijgt *niet* dezelfde kleur als punt 1. Omdat punt 3 ook niet dezelfde kleur mag krijgen als punt 2, zijn er nog $x - 2$ mogelijke kleuren over voor punt 3. Omdat punt 1 en punt 3 verschillende kleuren krijgen, zouden we deze twee punten dus net zo goed met elkaar kunnen verbinden.



In bovenstaand figuur staat een schematische weergave van de keuzemogelijkheden. Het chromatisch polynoom van graaf C is nu dus

$$\mathcal{P}_C(x) = x(x - 1) + x(x - 1)(x - 2) = x^3 - 2x^2 + x$$

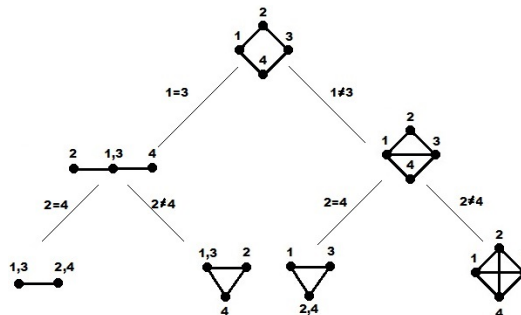
D (Zie figuur 13) Punt nummer 1 kan met x kleuren ingekleurd worden. Voor punt 2 zijn dan nog $x - 1$ kleuren over. Nu we bij punt 3 zijn aangekomen

zien we dat we ook hier twee keuzes hebben.

- (a) Punt 3 krijgt dezelfde kleur als punt 1. De punten kunnen nu samengevoegd worden. Een lijngraaf op drie punten blijft over. Voor punt 4 zijn nu ook weer twee mogelijkheden.
 - i. Punt 4 krijgt dezelfde kleur als punt 2. De punten kunnen samengevoegd worden en een volledige graaf op twee punten blijft over. Om deze graaf te kleuren zijn $x(x-1)$ mogelijkheden.
 - ii. Punt 4 krijgt niet dezelfde kleur als punt 2. De punten kunnen met elkaar verbonden worden en een volledige graaf op drie punten blijft over. Het aantal manieren om deze graaf te kleuren met x kleuren is $x(x-1)(x-2)$.
- (b) Punt 3 krijgt niet dezelfde kleur als punt 1. De punten kunnen dus verbonden worden met elkaar. Ook hier zijn weer twee mogelijkheden om punt 4 te kleuren.
 - i. Punt 4 krijgt dezelfde kleur als punt 2. De punten kunnen samengevoegd worden en er ontstaat een volledige graaf op drie punten die op $x(x-1)(x-2)$ manieren gekleurd kan worden.
 - ii. Punt 4 krijgt niet dezelfde kleur als punt 2. De punten kunnen met elkaar verbonden worden en er ontstaat een volledige graaf op 4 punten die op $x(x-1)(x-2)(x-3)$ manieren gekleurd kan worden.

Het chromatisch polynoom van graaf D is dus

$$\begin{aligned}
 \mathcal{P}_G &= x(x-1) + 2x(x-1)(x-2) + x(x-1)(x-2)(x-3) \\
 &= x^4 - 4x^3 + 6x^2 - 3x \\
 &= x(x-1)(x^2 - 3x + 3)
 \end{aligned}$$



Figuur 13: Kleuringen van graaf D .

Voor elke graaf G is het mogelijk om zo'n 'kleuringsboom' te maken. De boom is af als de grafen onderaan de boom (de bladeren) allemaal volledige grafen zijn. Voor de volledige grafen K_n is het gemakkelijk in te zien dat P_{K_n} een polynoom is. Er geldt namelijk dat:

$$P_{K_n}(x) = \frac{x!}{(x-n)!}$$

Het chromatisch polynoom van G is de optelsom van de chromatisch polynomen van alle bladeren, dus we zien nu dat P_G inderdaad een polynoom is voor elke graaf G .

Verder kan het aantal bladeren van deze kleuringsboom niet groter zijn dan $2^{\# \text{ kanten van } G^c}$, waarbij G^c het complement is van graaf $G = (V, E)$, ofwel $G^c = (V, (X \times X) \setminus E)$. Dit komt omdat we bij elke twee punten die nog niet verbonden zijn met elkaar, moeten kiezen of de punten dezelfde kleur of een verschillende kleur krijgen. Als ze dezelfde kleur krijgen voegen we de punten samen en als ze een verschillende kleur krijgen, verbinden we de punten. In totaal kunnen we er dus maximaal $\#(X \times X) \setminus E$ keer voor kiezen om twee punten samen te voegen. Dus het aantal bladeren van de kleuringsboom $\leq 2^{\#(X \times X) \setminus E}$.

Als we de kleuringsmethode toepassen op de sudokugraaf van orde 2, dan krijgen we het volgende chromatisch polynoom:

$$P(x) = x^{16} - 56x^{15} + 1492x^{14} - 25072x^{13} + 296918x^{12} - 2621552x^{11} + 17795572x^{10} - 94352168x^9 + 392779169x^8 - 1279118840x^7 + 3217758336x^6 - 6107865464x^5 + 8413745644x^4 - 7877463064x^3 + 4436831332x^2 - 1117762248x$$

Maar al voordat dit chromatisch polynoom daadwerkelijk bepaald was, kon er al het een en ander over gezegd worden:

Als we namelijk even terugdenken aan de sudoku van orde 2 en de bijbehorende sudokugraaf G , dan weten we dat voor $x = 0, 1, 2, 3$ moet gelden dat $P_G(x) = 0$, want elk punt van de sudokugraaf is bevat in meerdere K_4 -deelgrafen. Om precies te zijn is elk punt bevat in drie K_4 -grafen, namelijk die van de rij, kolom en het blok waarin het punt zich bevindt. En om K_4 op een juiste manier te kleuren heb je minstens 4 kleuren nodig. Dit is duidelijk aangezien alle Y_i in de sudoku 4 cellen bevatten en er dus minstens 4 symbolen of kleuren nodig zijn om de sudoku in te vullen of om de bijbehorende graaf te kleuren. Hieruit blijkt dat de bladeren van de kleuringsboom van de sudokugraaf slechts volledige grafen op ≥ 4 punten bevatten.

Opmerking 2.15. Naast het tellen van het aantal sudoku's, kun je de sudoku-grafen ook gebruiken om te bepalen of een sudokupuzzel een unieke oplossing heeft. Zo kun je namelijk het chromatisch polynoom geven van een graaf die al deels gekleurd is.

Het kleuren van de partieel gekleurde graaf, kun je zien als het oplossen van een sudokupuzzel.

Het is natuurlijk de bedoeling dat voor graaf G van een sudokupuzzel (dus G is al deels gekleurd) geldt dat $\mathcal{P}_G(4) = 4!$. Dat betekent namelijk dat de puzzel op kleurkeuze/symboolkeuze na, uniek oplosbaar is.

Verder weet je dat als $\mathcal{P}_G(4) = 0$, dat dan de sudokupuzzel in eerste instantie al niet juist gedefiniëerd was.

3 Structuur van de sudokupuzzel

Gegeven een sudoku, is het maken van een sudokupuzzel met die sudoku als oplossing, lang niet altijd even eenvoudig. Om een juiste verzameling clues te vinden, zou je een computerprogramma kunnen schrijven dat in een lege matrix steeds meer clues toevoegt, totdat de puzzel een unieke oplossing heeft (en dan eventueel nog een paar extra clues toevoegt). Je zou het programma ook andersom kunnen laten werken. Namelijk door te beginnen met een volledig ingevulde sudoku. Het programma kan dan voor één voor één symbolen uit de cellen weg laten en bij elke stap controleren of de dan overgebleven sudokupuzzel nog een unieke oplossing heeft. Dit soort algoritmes werken naar behoren, maar geven niet veel inzicht in de wiskundige theoriën van het maken van een sudokupuzzel. Dit hoofdstuk zal daar meer duidelijkheid over geven. Het volgende begrip zal daarbij veel gebruikt worden.

Definitie 3.1. Zij $\{X, \{Y_i\}_{i \in I}, \varphi\}$ een sudoku over S en zij $U \subseteq X$. U is een (*minimale*) *unavoidable set* van X als:

- $\exists \varphi' : X \rightarrow S$ met
 - $\forall x \in X \setminus U$ geldt $\varphi'(x) = \varphi(x)$
 - $\forall x \in U$: $\varphi'(x) \neq \varphi(x)$.
- $\{X, \{Y_i\}_{i \in I}, \varphi'\}$ is een sudoku over S

1	4	5	3	2	7	6	9	8
8	3	9	6	5	4	1	2	7
6	7	2	9	1	8	5	4	3
4	9	6	1	8	5	3	7	2
2	1	8	4	7	3	9	5	6
7	5	3	2	9	6	4	8	1
3	6	7	5	4	2	8	1	9
9	8	4	7	6	1	2	3	5
5	2	1	8	3	9	7	6	4

Figuur 14: Een unavoidable set

In de sudoku hierboven staat een voorbeeld van een unavoidable set. Als alle symbolen 1 een 2 worden en alle symbolen 2 een 1 worden, dan blijft

de matrix een sudoku. Als er bij deze sudoku een sudokupuzzel gemaakt moet worden, lijkt het dus zo te zijn dat ten minste één van deze cellen als clue gegeven moeten worden. Hierover gaat de volgende propositie.

Propositie 3.2. *Zij $\{X, \{Y_i\}_{i \in I}, Z, \varphi_0\}$ een sudokupuzzel met de sudoku $\{X, \{Y_i\}_{i \in I}, \varphi\}$ als oplossing en zij $\{U_j\}_{j \in J}$ de familie van alle unavoidable sets in de sudoku.*

Nu geldt dat $\{X, \{Y_i\}_{i \in I}, \varphi\}$ is de unieke oplossing van $\{X, \{Y_i\}_{i \in I}, Z, \varphi_0\} \Leftrightarrow \forall j \in J : Z \cap U_j \neq \emptyset$.

Bewijs. $(\Rightarrow)$: Stel $\exists j \in J$ met $U_j \cap Z = \emptyset$. Dan $\{X, \{Y_i\}_{i \in I}, \varphi'\}$ met $\varphi|_{X \setminus U_j} = \varphi'|_{X \setminus U_j}$ en $\varphi|_{U_j} \neq \varphi'|_{U_j}$ ook een sudoku. En dus heeft $\{X, \{Y_i\}_{i \in I}, Z, \varphi_0\}$ geen unieke oplossing.

Dus: $\{X, \{Y_i\}_{i \in I}, \varphi\}$ is de unieke oplossing van $\{X, \{Y_i\}_{i \in I}, Z, \varphi_0\} \Rightarrow \forall j \in J : Z \cap U_j \neq \emptyset$.

$(\Leftarrow)$: Stel $\{X, \{Y_i\}_{i \in I}, Z, \varphi_0\}$ heeft géén unieke oplossing. Dan zijn er dus sudoku's $\{X, \{Y_i\}_{i \in I}, \varphi\}$ en $\{X, \{Y_i\}_{i \in I}, \varphi'\}$ die beide oplossing zijn van de sudokupuzzel. Maar dan is $\{x \in X \mid \varphi(x) \neq \varphi'(x)\}$ een unavoidable set met $\{x \in X \mid \varphi(x) \neq \varphi'(x)\} \cap Z = \emptyset$. Dus nu zien we dat ook geldt dat: $\forall j \in J : Z \cap U_j \neq \emptyset \Rightarrow \{X, \{Y_i\}_{i \in I}, \varphi\}$ is de unieke oplossing van $\{X, \{Y_i\}_{i \in I}, Z, \varphi_0\}$. $\square$

Om het minimale aantal clues in een sudoku te bepalen, moet dus gezocht worden naar de minimale Z die alle U_i doorsnijdt. Dit klinkt echter makkelijker dan het is. Wel kan een simpelere eis afgeleid worden uit bovenstaande propositie.

Gevolg 3.3. $\#\varphi(Z) \geq \#S - 1$

Omdat voor alle s_1 en s_2 in S geldt dat $\{x \in X \mid \varphi(x) = s_1 \text{ of } \varphi(x) = s_2 \neq s_1\}$ een unavoidable set is, zien we dat er minstens $\#S - 1$ verschillende symbolen van de in totaal $\#S$ symbolen voor moeten komen in Z . Want alleen op die manier wordt elk van die unavoidable sets tenminste één keer geraakt.

Als sinds de eerst publicatie van een sudoku, zijn wetenschappers op zoek naar de zogenaamde minimale sudoku. Hiermee wordt een sudokupuzzel van orde 3 bedoeld met de kleinste verzameling clues. Dit probleem is een onderdeel van een hele familie problemen in de combinatoriek, waar al veel langer onderzoek naar wordt gedaan.

Al vrij snel werden er sudoku's gemaakt met slechts 17 clues, maar een sudoku met 16 clues werd niet gevonden. Jaren lang vermoedde men dat de minimale sudoku 17 clues heeft, maar een bewijs werd niet gevonden. Totdat Gary McGuire begin 2012 met een artikel kwam: *There is no 16-Clue*

Sudoku: Solving the Sudoku Minimum Number of Clues Problem. Hij heeft een slimme brute force methode bedacht om aan te tonen dat er geen geldige sudokupuzzel van orde 3 bestaat met slechts 16 clues. In zijn algoritme maakt hij gebruik van het volgende begrip:

Definitie 3.4. Zij $\{U_j\}_{j \in J}$ een verzameling unavoidable sets in een sudoku. Een *hitting set* van grootte m is $H \subseteq X$ zodat $\forall j \in J$ geldt dat $H \cap U_j \neq \emptyset$.

McGuire's programma werkt nu als volgt:

1. Maak een database met daarin alle 5472730538 essentieel verschillende sudoku's van orde 3.
2. Zoek per sudoku naar sudokupuzzels met 16 clues die de sudoku als unieke oplossing hebben. Doe dit als volgt:
 - (a) Vind een grote verzameling unavoidable sets in de sudoku.
 - (b) Vind alle hitting sets van grootte 16.
 - (c) Ga voor elke gevonden hitting set H na of $\{X, \{Y_i\}_{i \in I}, H, \varphi_0\}$ een geldige sudokupuzzel is met de oorspronkelijke sudoku als unieke oplossing.

In stap 2(a) wordt gezocht naar een grote verzameling unavoidable sets. Het vinden van alle unavoidable sets in een sudoku is niet nodig omdat het programma op zoek gaat naar alle mogelijke hitting sets van de gegeven familie unavoidable sets. Als er een hitting set is zodat $\{X, \{Y_i\}_{i \in I}, H, \varphi_0\}$ een juiste sudokupuzzel blijkt te zijn, dan volgt direct dat H een hitting set is van *alle* unavoidable sets.

Gezien het grote aantal essentieel verschillende sudoku's en het rekenwerk dat in dit algoritme te pas komt, heeft het toch een paar jaar geduurd voordat de conclusie getrokken kon worden dat er echt geen geldige sudokupuzzels van orde 3 zijn met slechts 16 clues.

4 Oplossingsstrategiën

In het volgende hoofdstuk zal een algoritme gegeven worden om de moeilijkheidsgraad van een gegeven sudoku te bepalen. Voordat we zo ver zijn, zullen we eerst wat meer inzicht moeten krijgen in de oplossingsstrategiën van de sudoku. De strategie die hieronder uitgelegd zal worden, is niet de strategie die door de meeste computerprogramma's gebruikt wordt. Dat is geen probleem omdat het ook de bedoeling is een moeilijkheidsgraad te definiëren die aangeeft hoe moeilijk het is om de sudoku met pen en papier op te lossen. We willen een wiskundige benadering geven van de menselijke manier van het oplossen van een sudoku. Daarvoor is het handig om een overzicht te hebben van welke symbolen er in welke cellen geplaatst zouden kunnen worden. We definiëren de mark-up van een cel.

Definitie 4.1. Stel dat φ_k al volledig gedefinieerd is en zij $Y_x = \bigcup Y_i$ met $x \in Y_i$. De *mark-up van een gegeven cel $x \in X$ in een sudokupuzzel* op dat moment is een $M_x^k \subseteq S$ zodat $M_x^k = S \setminus \bigcup_{y \in \text{dom}(\varphi_k|_{Y_x})} \{\varphi_k(y)\}$.

De mark-up van een cel x geeft dus precies aan welke elementen er op dat moment nog in x geplaatst kunnen worden zonder dat er direct een tegenspraak ontstaat. Alle symbolen die al in een Y_i staan waarvan x ook onderdeel is, komen dus niet voor in de mark-up van x . In figuur 15 bekijken we de mark-up van de rode cel. In de rij waarin de rode cel, x , zich bevindt, is er al een 1 ingevuld en in zowel de kolom als in het blok waarin x zich bevindt, staat al een 2. We zien nu dus dat $M_x^k = \{3, 4\}$.

	1		34
		2	
3			
			2

Figuur 15: Mark-up van de rode cel

Er zijn sudoku's waarin een groot deel van de cellen een mark-up heeft bestaande uit één symbool en zo dat als je die getallen invult, de mark-ups van de overige cellen zich ook zullen beperken tot één symbool. De sudoku is dan snel en gemakkelijk op te lossen, maar in het algemeen zal dit niet het geval zijn en is er wat meer denkwerk vereist. In die gevallen zullen we *pre-emptive sets* nodig hebben. We geven eerst een voorlopige definitie die inzicht geeft in het begrip. Dan volgen een aantal proposities, waarna een exacte definitie van een pre-emptive set gegeven wordt.

Definitie 4.2. (Voorlopig)

Een *pre-emptive set van orde m* is een paar (P, S_P) waarbij $P \subseteq Y_i$ voor een $i \in I$ en $S_P \subseteq S$ met $\#P = \#S_P$ waarvoor je op grond van de mark-up en de al ingevulde clues al weet dat $\varphi(P) = S_P$ moet gelden.

Opmerking 4.3. Voor alle i in I is (Y_i, S) een pre-emptive set van orde $\#S$ en voor alle z in Z is $(z, \varphi_0(z))$ een pre-emptive set van orde 1.

Wanneer er een pre-emptive set van orde 1 wordt gevonden, dus als $M_x^k = \{s\}$ of als s uitsluitend voorkomt in de mark-up van x , dan is het duidelijk dat voor φ moet gelden dat $\varphi(x) = s$. Dit verschijnsel kunnen we nu definiëren in termen van pre-emptive sets. En wel op de volgende manier.

Definitie 4.4. Een *geforceerd symbol* is een pre-emptive set van orde 1.

Tijdens de zoektocht naar pre-emptive sets in een sudoku zullen vele pre-emptive sets naar voren komen. Het merendeel daarvan is echter triviaal. Een pre-emptive set (P, S_P) is natuurlijk triviaal als deze van orde $\#S$ is, maar ook als P precies alle niet ingevulde cellen in de bijbehorende Y_i bevat en als S_P precies alle nog niet-gebruikte symbolen bevat.

Hieronder een klein voorbeeldje van een pre-emptive set. In de figuur hieronder is een rij (een Y_i) uit een sudokupuzzel te zien, met daarin twee gegeven clues in de tweede cel, $c(2)$, en in de laatste, $c(9)$.

134	8	579	1347	4679	57	45679	479	2
----------------	----------	----------------	-----------------	-----------------	---------------	------------------	----------------	----------

Figuur 16: Voorbeeld van pre-emptive sets

De symbolen 1 en 3 komen alleen in $c(1)$ en in $c(4)$ voor en zullen dus uiteindelijk in die twee cellen geplaatst moeten worden. $(\{c(1), c(4)\}, \{1, 3\})$ is dus een pre-emptive set. Verder komen in de mark-ups van de cellen $c(3)$, $c(5)$, $c(6)$, $c(7)$ en $c(8)$ alleen de symbolen 4, 5, 6, 7 en 9 voor, dus $(\{c(3), c(5), c(6), c(7), c(8)\}, \{4, 5, 6, 7, 9\})$ is ook een pre-emptive set.

We zien nu dat er twee soorten pre-emptive sets zijn:

1. $\bigcup_{x \in P} M_x = S_P$, oftewel: In de mark-up van de cellen van P_i komen alleen maar getallen uit S_P voor.
2. $\bigcup_{x \in Y_i \setminus P} M_x = S \setminus S_P$, oftewel: De getallen uit S_P komen niet voor in de mark-up van de cellen van $Y_i \setminus P$.

Propositie 4.5. *Een pre-emptive set is altijd van één van bovenstaande soorten.*

Bewijs. Stel (P, S_P) is een pre-emptive set van orde $\#P$. Dan moet het per definitie zo zijn dat de elementen uit S_P in de cellen uit P geplaatst moeten worden. Het is dus ook genoeg om te laten zien dat de elementen uit S_P niet in de cellen uit $Y_i \setminus P$ geplaatst kunnen worden.

Dit is vanzelfsprekend in het tweede geval omdat de elementen uit S_P niet voor komen in de mark-up van de cellen van $Y_i \setminus P$.

Stel nu dat (P, S_P) een pre-emptive set is die niet van de tweede soort is.

Dan is er een $s \in S_P$ is zodat s wel in $\bigcup_{x \in Y_i \setminus P} M_x$ voorkomt. En dus zou je s in een van de cellen uit $Y_i \setminus P$ kunnen plaatsen. Maar omdat $\#P = \#S_P$, zien we nu dat er een $t \in S \setminus S_P$ in één van de cellen uit P geplaatst moet worden. Maar dat zou betekenen dat (P, S_P) geen pre-emptive set is, wat in tegenspraak is met de aanname.

Omdat er een $s \in S_P$ wel in $\bigcup_{x \in Y_i \setminus P} M_x$ voorkomt, kunnen we nu concluderen dat er geen $s \in S \setminus S_P$ in $\bigcup_{x \in P} M_x$ voor mag komen en dus is het een pre-emptive set van de eerste soort. $\square$

Wat nu opvalt is dat als er een pre-emptive set gevonden wordt, dat er in die Y_i dan altijd nog een pre-emptive set te vinden is. De pre-emptive sets komen dus in paren. We geven nu een exacte definitie van een paar van pre-emptive sets:

Definitie 4.6. Zij $J \neq \emptyset$ een indexverzameling en voor $P \subseteq Y_i$ schrijf $P^c = Y_i \setminus P$.

$\{(P, S_P), (P^c, S_{P^c})\}$ is een paar van pre-emptive sets

$\Leftrightarrow$

- $\exists i \in I$ zodat $P \subseteq Y_i$
- $\exists A \in \{P, P^c\}$ waarvoor geldt dat er niet-lege $B_j \subseteq S$ zijn zodat $B_j \cap \bigcup_{x \in Y_i \setminus A} M_x^k = \emptyset$. Oftewel, voor een $A \in \{P, P^c\}$ geldt dat de elementen uit $S_{Y_i \setminus A}$ niet in de mark-up van de cellen uit A voorkomen.
Dan $S_A = \bigcup_{j \in J} B_j$ en $S_{Y_i \setminus A} = S \setminus S_A$
- $\#A = \#S_A$

Opmerking 4.7. (A, S_A) is een pre-emptive set van orde $\#A$.

Propositie 4.8. Zij (P, S_P) een pre-emptive set van de ene soort van orde $\leq \#S - 1$. Dan is (P^c, S_{P^c}) , met $P^c = Y_i \setminus P$, een pre-emptive set van de andere soort van orde $\leq \#S - 1$.

Bewijs. (P, S_P) is een pre-emptive set van de eerste soort, dus $\bigcup_{x \in P} M_x = S_P$.

De elementen uit $S \setminus S_P = S_{P^c}$, komen nu niet voor in $\bigcup_{x \in P} M_x = \bigcup_{x \in Y_i \setminus P^c} M_x$ en daarmee zien we dat $\bigcup_{x \in Y_i \setminus P^c} M_x = S \setminus S_{P^c}$ dus is (P^c, S_{P^c}) een pre-emptive set van de tweede soort.

Zij nu (P, S_P) een pre-emptive set van de tweede soort, dus $\bigcup_{x \in Y_i \setminus P} M_x = S \setminus S_P$.

Dus nu zien we dat $\bigcup_{x \in P^c} M_x = \bigcup_{x \in Y_i \setminus P} M_x = S \setminus S_P = S_{P^c}$ en dus is (P^c, S_{P^c}) een pre-emptive set van de eerste soort. $\square$

Er is nu al een hoop verteld over pre-emptive sets, maar hoe zijn deze pre-emptive sets nu te gebruiken bij het oplossen van een sudoku?

Het *verwerken van een pre-emptive set* gaat als volgt:

1. Als er een pre-emptive set van orde 1 wordt gevonden, zeg (x, s) , dan weten we dus dat voor φ moet gelden dat $\varphi(x) = s$ en dus kunnen we s invullen in x . Verder kunnen we nu x wegstrepen uit de mark-ups van alle cellen y in alle Y_i waarvoor geldt dat $x \in Y_i$.
2. Als er een pre-emptive set van orde $m \geq 2$ wordt gevonden, zeg $(P, S_P) = (\{x_1, \dots, x_m\}, \{s_1, \dots, s_m\})$, dan kunnen we nog niet concreet een getal gaan invullen, want we weten nog niet hoe $\varphi \upharpoonright_P$ precies gedefinieerd is. Wat we wél kunnen doen, is alle $s \in S_P$ wegstrepen uit de mark-ups van de cellen uit $Y_i \setminus P$, in het geval dat (P, S_P) een pre-emptive set van de eerste soort is. En in de mark-ups van de cellen $x \in P$ kunnen we de elementen uit $S \setminus S_P$ wegstrepen, in het geval dat (P, S_P) een pre-emptive set van de tweede soort is.

Door het verwerken van pre-emptive sets, kun je dus mark-ups van andere cellen verkleinen waardoor mogelijk weer nieuwe pre-emptive sets ontstaan. Hoe meer pre-emptive sets je dus verwerkt, hoe meer forced symbols te voorschijn komen en hoe preciezer je φ kunt definiëren, tot uiteindelijk de oplossing tevoorschijn komt.

Propositie 4.9. *Als je de sudokupuzzel op kunt lossen uitsluitend door pre-emptive sets te verwerken, dan heeft de sudokupuzzel een unieke oplossing.*

Bewijs. Met het verwerken van pre-emptive sets, worden één voor één de elementen uit de mark-up geëlimineerd. Er wordt pas een symbool ingevuld (vastgesteld dat $\varphi(x) = s$) als blijkt dat $(\{x\}, \{s\})$ een pre-emptive set is. Dan ligt voor die cel x dus vast dat alleen s ingevuld kan worden. Als er een oplossing wordt gevonden enkel door pre-emptive sets te verwerken, dan is φ dus uniek bepaald. $\square$

Het omgekeerde is echter niet waar. Arto Inkala heeft een sudoku gemaakt die uniek oplosbaar is en toch niet op te lossen is door uitsluitend pre-emptive sets te verwerken.

8								
		3	6					
	7			9		2		
	5				7			
				4	5	7		
			1				3	
		1					6	8
		8	5				1	
	9					4		

Figuur 17: Arto Inkala's sudoku

Voor dit soort puzzels zal dus een andere oplossingsstrategie bepaald moeten worden. De methode die we gebruiken is backtracken en werkt als volgt. Omdat er op grond van pre-emptive sets, geen nieuwe φ_k gedefinieerd kan worden, kiezen we een willekeurige nog lege cel x en definiëren $\varphi_k(x) = s \in M_x^{k-1}$. Vervolgens lossen we de puzzel verder op tot er een oplossing gevonden is of tot er een tegenspraak gevonden wordt, bijvoorbeeld dat er een $i \in I$ is zodat $\varphi' \upharpoonright_{Y_i}$ niet bijectief is. Als er een tegenspraak gevonden wordt, dan was de keuze $\varphi_k(x) = s$ niet juist en kiezen we hiervoor een andere $s \in M_x^{k-1}$.

Definitie 4.10. Zij φ_{k-1} de laatst gedefinieerde φ_i . En stel dat er geen niet-triviale pre-emptive sets meer te vinden zijn in de puzzel. *Backtracken* werkt als volgt:

1. Kies een $x \in X \setminus \text{dom}(\varphi_{k-1})$.
2. Definieer φ' zodat $\forall i$ met $0 \leq i \leq k-1$ geldt dat $\varphi_i = \varphi'_i$ en definieer $\varphi'_k(x) = s$ voor een willekeurige $s \in M_x^{k-1}$.
 - Als er geen tegenspraak gevonden wordt, definieer dan $\varphi = \varphi'$ zodat $\{X, \{Y_i\}_{i \in I}, \varphi\}$ de oplossing van de puzzel is.

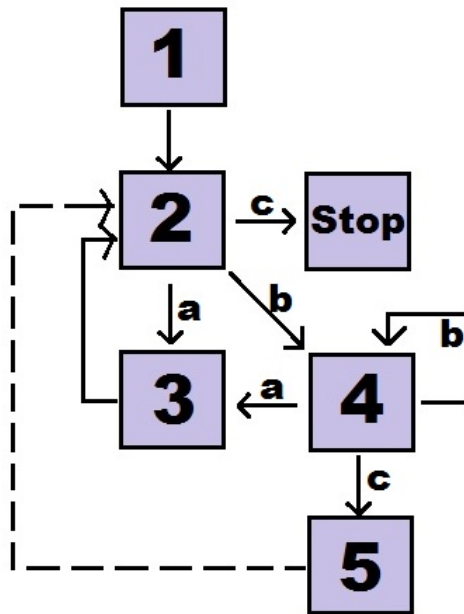
- Als er wel een tegenspraak gevonden wordt, vergeet dan φ' ga dan terug naar stap 2 en definieer $\varphi'_k(x) = t \neq s$ voor een andere $t \in M_x^{k-1}$.

Bijna alle computerprogramma's die sudokupuzzels oplossen maken gebruik van, of zijn zelfs geheel gebaseerd, op deze techniek. Wanneer je een sudoku echter met pen en papier oplost, zal men in het algemeen deze techniek juist willen vermijden omdat het erg tijdrovend is en je vaak meerdere pogingen moet doen voordat je bij de juiste oplossing terecht komt.

Het volgende algoritme zal gebruikt worden voor het oplossen van $\{X, Z, \{Y_i\}_{i \in I}, \varphi_0\}$, een gegeven sudokupuzzel over S .

1. $\forall x \in X \setminus Z$ bepaal M_x^k . Vervolg met stap 2.
2. Vind een pre-emptive set van orde 1 (de forced symbols).
 - (a) Forced symbol gevonden? Vervolg met stap 3.
 - (b) Geen forced symbols meer in de puzzel én $\varphi_{\#X-\#Z}$ nog niet gedefinieerd? Vervolg met stap 4.
 - (c) Geen forced symbols meer in de puzzel én $\varphi_{\#X-\#Z}$ is volledig gedefinieerd? De puzzel is opgelost en $\varphi_{\#X-\#Z} = \varphi$
3. Verwerk de gevonden pre-emptive sets en de bijbehorende complement sets. Als (x, s) een forced symbol is en φ_{k-1} de laatst gedefinieerde φ_i , definieer dan $\varphi_k(x) = s$. Vervolg met stap 2.
4. Kies $i \in I$ willekeurig en zoek een pre-emptive set (P, S_P) met $P \subseteq Y_i$ en $\#P \leq 5$.
 - (a) Pre-emptive set gevonden? Vervolg met stap 3.
 - (b) Geen pre-emptive set gevonden? Herhaal stap 4 met een andere willekeurige $i \in I$.
 - (c) Geen pre-emptive set gevonden $\forall i \in I$? Vervolg met stap 5.
5. Backtracken zoals beschreven in de definitie. Ga verder met stap 2, maar gebruik vanaf nu φ' in plaats van φ .
 - (a) Wordt er geen tegenspraak gevonden? De puzzel is opgelost en $\varphi'_{\#X-\#Z} = \varphi$
 - (b) Wordt er een tegenspraak gevonden? φ' wordt gewist en stap 5 wordt herhaald met een anders gekozen $\varphi_k(x)$.

Het figuur hieronder geeft een schematische weergave van het algoritme.



Figuur 18: Schematische weergave van het oplossingsalgoritme.

Opmerking 4.11. • Bij het verwerken van de pre-emptive sets, hoort natuurlijk ook het aanpassen van de mark-ups van alle cellen in de bijbehorende Y_i .

- Bij stap 4.a is het voldoende om te zoeken naar pre-emptive sets van orde ≤ 5 omdat bij stap 4.b de complementen van de gevonden pre-emptive sets ook direct verwerkt worden. Zo worden dus ook pre-emptive sets van orde ≥ 5 verwerkt.
- Er zou ook voor gekozen kunnen worden om eerst pre-emptive sets te zoeken van orde 2, dan van orde 3 etc. in plaats van willekeurige orde. Eerst pre-emptive sets van kleinere orde zoeken zou handig zijn omdat die pre-emptive sets in het algemeen gemakkelijker te herkennen zijn. Toch is het verstandig om naar pre-emptive sets van willekeurige orde te zoeken omdat het bij het oplossen van een sudoku ook vaak voor komt dat er eerst een pre-emptive set van grotere orde gevonden wordt. Het is dan onzinnig om niks te doen met die set en op zoek te gaan naar een set van kleinere orde.

5 De moeilijkheidsgraad bepalen

De moeilijkheidsgraad van een gegeven sudokupuzzel bepalen, doen we aan de hand van het oplossingsalgoritme. We kunnen nu namelijk voor elke φ_i noteren welke preemptive sets allemaal zijn verwerkt na het definiëren van φ_{i-1} . Oftewel, voor elke nieuwe cel die ingevuld wordt, bepalen we welke stappen ondernomen zijn om die cel in te vullen. Vervolgens wordt voor elke genomen stap een bepaald aantal punten bij de moeilijkheidsgraad opgeteld.

Om de moeilijkheidsgraad van sudokupuzzel $\{X, Z, \{Y_i\}_{i \in I}, \varphi_0\}$ te bepalen gebruiken we de volgende formule:

$$M = \sum_{k \in \{\#X - \#Z, \dots, \#X\}} w(x_k)$$

Waarbij

- $x_k = \text{dom}(\varphi_k) \setminus \text{dom}(\varphi_{k-1})$ oftewel x_k is de k -de cel die ingevuld wordt
-

$$w(x_k) = \begin{cases} \prod_{i \in \{1, \dots, n\}} m_i & \text{in geval 1} \\ (\#X - \#Z - k)^2 + \prod_{i \in \{1, \dots, n\}} m_i & \text{in geval 2} \end{cases}$$

- Geval 1: Als $(x_k, \varphi_k(x_k))$ vastgelegd nadat er n pre-emptive sets van orde m_i zijn opgelost.
- Geval 2: Als $(x_k, \varphi_k(x_k))$ is vastgelegd in de backtrack-stap, dus als $(x_k, \varphi_k(x_k))$ is gekozen zonder dat $(x_k, \varphi_k(x_k))$ een forced symbol was.

Opmerking 5.1. • Als een sudokupuzzel oplosbaar is door slechts forced symbols te verwerken, dan krijgt deze puzzel dus als moeilijkheidsgraad $\#X - \#Z$.

- Omdat er in het oplossingsalgoritme pas gebacktrackt wordt als er echt geen pre-emptive set meer gevonden wordt, zien we dus dat backtracken altijd op hetzelfde moment zal gebeuren. Het kan niet zo zijn dat er de ene keer eerder gebacktrackt moet worden dan de andere keer.
- Als er bij het backtracken een tegenspraak wordt veroorzaakt, worden alle beslissingen in het backtrackproces ongedaan gemaakt, maar de waarden die bij de moeilijkheidsgraad zijn opgeteld, blijven wel. Dit

is realistisch omdat een puzzel moeilijker is als je vaak een verkeerde beslissing maakt.

Omdat er in het oplossingsalgoritme een aantal keer een willekeurige $\{Y_i\}$ wordt gekozen, zal de moeilijkheidsgraad niet elke keer precies hetzelfde zijn. Voor dit probleem zijn verschillende oplossingen.

1. Er zou een boom gemaakt kunnen worden van alle mogelijke manieren om de sudokupuzzel op te lossen. Vervolgens zou per tak de moeilijkheidsgraad bepaalt kunnen worden. Als moeilijkheidsgraad wordt dan het gemiddelde over de moeilijkheidsgraden van alle takken genomen.
2. Op dezelfde manier al bij het vorige punt zou een boom van oplossingen gemaakt kunnen worden. In plaats van te middelen over alle moeilijkheidsgraden, zou ook het maximum of juist het minimum als moeilijkheidsgraad genomen kunnen worden.
3. Omdat er soms heel veel verschillende manieren zijn om een sudokupuzzel op te lossen, kan het maken van zo'n boom erg veel tijd in kosten. In plaats van alle mogelijke oplossingen bepalen, zou je het algoritme ook bijvoorbeeld 10 keer kunnen laten lopen om vervolgens de moeilijkheidsgraden van deze 10 manieren van oplossen te middelen.
4. Omdat het niet zeker is dat 10 verschillende oplossingsmethodes een goed beeld geeft van hoe moeilijk het gemiddeld is om de sudokupuzzel op te lossen, zou je ook net zo lang het algoritme kunnen laten lopen (en steeds de gemiddelde moeilijkheidsgraad bepalen) totdat het gemiddelde ongeveer constant blijft.

Oplossing 1 en 2 zullen erg veel tijd in beslag nemen dus lijken 3 en 4 een betere oplossing. Daarnaast kan oplossing 2 misleidend zijn omdat het maar zelden voor zal komen dat de puzzel inderdaad opgelost wordt via de moeilijkste of juist via de makkelijkste weg. Vaak zal het er toch ergens tussenin zitten. Theoretisch gezien lijkt punt 4 de beste oplossing, maar omdat de oplossingsmogelijkheden van een puzzel misschien wel erg variëren, zal punt 4 ook erg veel tijd in beslag kunnen nemen, waardoor punt 3 in de praktijk de beste oplossing lijkt te zijn.

Opmerking 5.2. De moeilijkheidsgraad zou natuurlijk ook op een aantal andere manieren bepaald kunnen worden. Je zou bijvoorbeeld kunnen kijken naar de gegeven clues. Bij makkelijke sudokupuzzels is het zo vaak zo dat je meerdere clues weg kunt laten en wel zo dat de puzzel nog steeds een unieke

oplossing heeft. Als vanzelfsprekend kan dat niet bij een sudokupuzzel met slechts 17 clues. Maar er zijn ook sudokupuzzels met meer dan 17 clues waarbij het weglaten van een clue toch resulteert in een sudokupuzzel met meerdere oplossingen.

Het aantal oplossing dat ontstaat bij het weglaten van een clue hangt samen met de unavoidable sets. Bij een sudokupuzzel met 17 clues, zijn de clues op de meest strategische manier geplaatst. Daarmee wordt bedoeld dat elke clue hoogstwaarschijnlijk meerdere unavoidable sets raakt die nog niet door andere clues geraakt zijn. Als deze clue dus weg wordt gelaten, zullen meerdere van deze unavoidable sets niet meer geraakt worden, waardoor de puzzel vele verschillende oplossingen zal hebben. In de eerdergenoemde puzzel van Arto Inkala is het zo dat als je één clue weg zou laten, dat de puzzel dan gemiddeld 145600 verschillende oplossingen heeft in plaats van één oplossing!

Referenties

- [1] Van den Essen, Arno (2006).
Magische vierkanten: Van Lo-Shu tot sudoku.
De wonderbaarlijke geschiedenis van wiskundige puzzels. 1e druk, hoofdstuk 11.
- [2] Herzberg, Agnes & Murty, M. Ram (2007).
Sudoku Squares and Chromatic Polynomials
- [3] Felgenhauer, Bertram & Jarvis, Frazer (2005).
Enumerating possible Sudoku grids
- [4] Russell, Ed & Jarvis, Frazer (2005).
There are 5472730538 essentially different Sudoku grids ... and the Sudoku symmetry group
- [5] McGuire, Gary (2012).
There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem.