

Anti-magische grafen

Joris Sprunken

Studentnummer: 0212725

Bachelor opleiding wiskunde

**Radboud Universiteit
Nijmegen**



Begeleider: Dr. W. Bosma
Tweede corrector: Dr. B. Souvignier

Inhoudsopgave

0	Notaties en Basisdefinities	1
1	Inleiding	3
2	Voorbeelden van anti-magische families	5
2.1	<i>Het pad: P_n</i>	5
2.2	<i>De cykel: C_n</i>	5
2.3	<i>Het wiel: W_n</i>	6
2.4	<i>De volledige graaf: K_n</i>	7
2.5	<i>Dichte grafen</i>	8
2.6	<i>Verdere bekende resultaten</i>	9
3	Het genereren van bomen	10
3.1	<i>Alle gerichte bossen genereren</i>	10
3.2	<i>Implementatie in Magma</i>	12
3.3	<i>Het genereren van alle vrije bomen</i>	13
3.4	<i>Implementatie in Magma</i>	15
4	Het vinden van anti-magische labelingen	17
4.1	<i>De raamfunctie: ‘Testallebomen’</i>	17
4.1.1	<i>Implementatie in Magma</i>	17
4.2	<i>Hulpfunctie 1: ‘VertaalNaarMatrix’</i>	18
4.2.1	<i>Implementatie in Magma</i>	18
4.3	<i>Hulpfunctie 2: ‘Puntsom’</i>	18
4.3.1	<i>Implementatie in Magma</i>	19
4.4	<i>Hulpfunctie 3: ‘VolgendeRijtje’</i>	19
4.4.1	<i>Implementatie in Magma</i>	20
4.5	<i>De hoofdfunctie: ‘Antimagischboom’</i>	20
4.5.1	<i>Implementatie in Magma</i>	21
5	Resultaten, verbeterpunten en verdere zoekrichtingen	22
5.1	<i>Resultaten</i>	22
5.2	<i>Verbeterpunten</i>	25
5.3	<i>Verdere zoekrichtingen</i>	25
6	Slot	27

0 Notaties en Basisdefinities

In deze scriptie zullen we enkele basisbegrippen uit de grafentheorie veel gebruiken. Enkele hiervan worden hier gedefinieerd en er wordt een notatie afgesproken.

Een graaf zullen we noteren als $G(V_G, E_G)$, waarbij V_G de punten van de graaf voorstellen en E_G de kanten van de graaf. Een graaf heeft altijd eindig veel punten en kanten: ($V_G < \infty$, $E_G < \infty$). Wanneer duidelijk is over welke graaf we het hebben, zullen we de verzameling van punten kanten ook wel noteren als V , respectievelijk E . De graaf zullen we ook wel noteren als G . Punten zullen we noteren als x , een kant tussen x en y als $\langle x, y \rangle$ (merk op dat er tussen twee gegeven punten hoogstens één kant loopt).

Wanneer we spreken over grafen, bedoelen we dat de kanten niet gericht zijn (1) en dat er geen lussen zijn (2). Formeel gezegd is E een irreflexieve, symmetrische relatie op V , oftewel:

1. $\forall G \forall x, y \in V_G [\langle x, y \rangle \in E_G \Leftrightarrow \langle y, x \rangle \in E_G]$;
2. $\forall G \forall x \in V_G [\langle x, x \rangle \notin E_G]$.

Bij elk punt x uit V_G definiëren we de verzameling van zijn burens, $B(x)$ als $\{y : \langle x, y \rangle \in E_G\}$. Het aantal elementen van $B(x)$ noemen we de graad van x : $d(x)$.

We kunnen nu ook bij elk punt x de verzameling B^n van burens op afstand n recursief definiëren door:

$$\begin{aligned} B^1(x) &= B(x) \\ B^n(x) &= \bigcup_{y \in B^{n-1}(x)} B(y). \end{aligned}$$

We noemen een graaf G samenhangend als

$$\forall x \in V_G \exists n \forall y \in V_G [y \neq x \rightarrow y \in B^n(x)].$$

Een boom $B(V_B, E_B)$ met n punten is een samenhangende graaf met $n - 1$ kanten.

Een bos F is een eindige verzameling $\{f_1, f_2, \dots, f_n\}$ waarbij f_i een boom is voor alle $1 \leq i \leq n$. Het aantal punten van het bos is $\sum_{i=1}^n \#(V_{f_i})$.

We noemen twee grafen G en H isomorf als ze beide evenveel (p) punten hebben en er injectieve afbeeldingen $N : \{1, \dots, p\} \rightarrow V_G$ en $M : \{1, \dots, p\} \rightarrow V_H$ bestaan zodat $\forall a, b \leq p [\langle N(a), N(b) \rangle \in E_G \Leftrightarrow \langle M(a), M(b) \rangle \in E_H]$. We zullen dit ook wel noteren als $G \cong H$.

Twee bossen $F_1 = \{f_1, \dots, f_n\}$ en $F_2 = \{g_1, \dots, g_n\}$ heten isomorf als er een injectieve afbeelding $N : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ bestaat, zodat $\forall i \leq n [f_i \cong g_{N(i)}]$.

Bij een gegeven samenhangende graaf G , definiëren een afstandsfunctie $Dist : V_G \times V_G \rightarrow \{0, \dots, \#V_G\}$ door:

$$\begin{aligned} Dist(x, x) &= 0 \\ Dist(x, y) &= n \quad \text{als} \quad y \in B^n(x) \wedge \forall m [m < n \rightarrow y \notin B^m]. \end{aligned}$$

Bij een boom B definiëren we de verzameling C_B van centroiden als:

$$\{x : x \in V_B \mid \forall y \in V_B [\sum_{z \in V_B} Dist(x, z) \leq \sum_{z \in V_B} Dist(y, z)]\}.$$

Wanneer we een waarde aan een kant $\langle x, y \rangle$ toe willen voegen, geven we deze waarde weer als $W_{\langle x, y \rangle}$.

Overige notaties die we vaak zullen gebruiken, is p voor het aantal punten ($\#V$) en q voor het aantal kanten van de graaf ($\#E$).

1 Inleiding

Al vele duizenden jaren is de mensheid gefascineerd door getallenpatronen in de wereld om ons heen. Een wereldberoemd voorbeeld hiervan is het magisch vierkant van Lo-Shu. Door deze fascinatie is er over de gehele wereld een grote groep, ietwat oneerbiedig zogenoemde, ‘recreatieve wiskundigen’ ontstaan.

Ook binnen de grafentheorie zijn hier talloze voorbeelden van te vinden, zoals de generaliseerde versie van het magisch vierkant: de magische graaf.

Definitie: Een graaf G heet magisch als er een eeneenduidige functie $W : E_G \rightarrow \{1, \dots, q\}$ bestaat, zodanig dat voor elke x, y in V_G :

$$\sum_{a \in B(x)} W(\langle x, a \rangle) = \sum_{b \in B(y)} W(\langle y, b \rangle).$$

Zulk een functie noemen we ook wel een magische labeling.

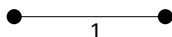
In het vervolg van deze scriptie zullen we kijken naar grafen die juist anti-magisch gelabeld kunnen worden. Na de vorige definitie is het weinig verrassend dat dit zal betekenen dat de som van de functiewaarden bij elk punt juist verschillend moeten zijn, oftewel:

Definitie: Een graaf G heet anti-magisch als er een eeneenduidige functie $W : E_G \rightarrow \{1, \dots, q\}$ bestaat, zodanig dat voor elke x, y ($x \neq y$) in V_G :

$$\sum_{a \in B(x)} W(\langle x, a \rangle) \neq \sum_{b \in B(y)} W(\langle y, b \rangle).$$

$\sum_{a \in B(x)} W(\langle x, a \rangle)$ zullen we ook wel noteren als $S(x)$.

Geheel analoog zullen we zulk een functie een anti-magische labeling noemen.



Hierboven is een voorbeeld te zien van een labeling die niet anti-magisch is; immers, bij elk van beide punten in deze graaf (ook wel K_2 genoemd), is de som van de waarden van alle aangrenzende kanten gelijk aan 1. Het is duidelijk dat dit de enige eeneenduidige afbeelding van E_G naar $\{1\}$ is, dus is K_2 niet anti-magisch te labelen.

Hieronder zien we de graaf P_2 met een labeling W van de kanten:



Als we de punten van links naar rechts aanduiden met a , b en c , dan zien we dat $W(\langle a, b \rangle) = 1$ en $W(\langle b, c \rangle) = 2$, dus dat $\sum_{x \in B(a)} W(\langle a, x \rangle) = 1$, $\sum_{x \in B(b)} W(\langle b, x \rangle) = 3$, en $\sum_{x \in B(c)} W(\langle c, x \rangle) = 2$. De som van de waarden is dus bij elk punt verschillend en dus is W een anti-magische labeling van de graaf P_2 .

We hebben nu gezien dat er grafen bestaan die anti-magisch zijn en grafen die niet anti-magisch gelabeld kunnen worden. Een natuurlijke en interessante vraag die zich nu opwerpt, is: “Welke grafen zijn anti-magisch?” In [1] wordt hierop als voorzichtig antwoord het vermoeden uitgesproken “Alle samenhangende grafen, behalve de K_2 zijn anti-magisch”. Tot op dit moment is er nog geen bewijs, noch een tegen-

voorbeeld bekend voor dit vermoeden, evenals voor het deelantwoord “Alle bomen, behalve de K_2 zijn antimagisch”.

Voor het schrijven van deze scriptie heb ik onderzoek gedaan naar het antwoord op deze vragen. In hoofdstuk twee zal ik enkele families van grafen geven met een algemene methode om een anti-magische labeling te construeren, in hoofdstuk drie zal ik laten zien hoe ik het computeralgebrapakket Magma (zie [4]) gebruikt heb om bomen te genereren, waarna ik in hoofdstuk vier algoritmen beschrijf die ik heb gebruikt om een vermoeden te krijgen over bomen die ‘moeilijk’ te labelen zijn. In hoofdstuk vijf zal ik de resultaten van deze zoektocht geven, samen met enkele verbeterpunten en verdere zoekrichtingen. Tot slot maak ik in hoofdstuk zes nog enkele afsluitende opmerkingen.

2 Voorbeelden van anti-magische families

Voordat we gaan proberen iets over alle grafen, of over alle bomen te zeggen, is het fijn om een gevoel te krijgen voor anti-magische labelingen. Daarom zal ik in dit hoofdstuk enkele bekende families van grafen introduceren, samen met een methode om een anti-magische labeling te maken voor deze families.

2.1 Het pad: P_n

Het pad van lengte $n > 1$, P_n , is een graaf met $n + 1$ punten en n kanten, zodat er twee punten b en e zijn met $d(b) = d(e) = 1$ en voor de $n - 1$ punten $m_1, \dots, m_{n-1}$ ongelijk aan b en e geldt $d(m_i) = 2$. Dit betekent dat we de punten van P_n kunnen nummeren van $1, \dots, n + 1$, zodat $E_{P_n} = \{\langle i, i + 1 \rangle : i \in \{1, \dots, n\}\}$. Hieronder is een afbeelding gegeven van de P_6 :



We zullen nu aantonen dat de P_n anti-magisch is voor alle n . Hierbij maken we onderscheid tussen even en oneven n :

- Als n even is, dan is de volgende afbeelding een anti-magische labeling:

$$W(\langle i, i + 1 \rangle) = i \quad (i \in \{1, \dots, n\}).$$

Bewijs: We zien dat $S(n + 1) = n$. Voor alle $i \in \{1, \dots, n\}$ geldt

$S(i) = i + (i - 1) = 2i - 1$. Duidelijk is dat voor $i \neq j$ geldt $2i - 1 \neq 2j - 1$ en omdat $S(i) = 2i - 1$ oneven is voor $i \in \{1, \dots, n\}$, geldt $S(i) \neq n = S(n + 1)$.

- Als n oneven is, dan is de volgende afbeelding een anti-magische labeling:

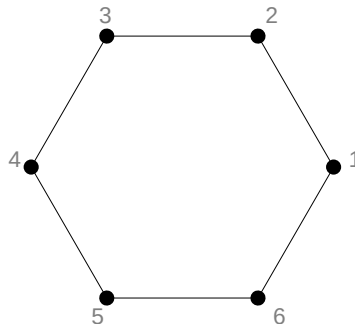
$$W(\langle i, i + 1 \rangle) = i + 1 \quad (i \in \{1, \dots, n - 1\})$$

$$W(\langle n, n + 1 \rangle) = 1.$$

Bewijs: We zien dat $S(1) = 2$, $S(n + 1) = 1$ en $S(n) = n + 1 > 2$. Voor alle $i \in \{2, \dots, n - 1\}$ geldt $S(i) = i + (i + 1) = 2i + 1 > 1$. Duidelijk is dat voor $i \neq j$ geldt $2i + 1 \neq 2j + 1$ en omdat $S(i) = 2i - 1$ oneven is voor $i \in \{1, \dots, n\}$, geldt $S(i) \neq 2 = S(1)$ en $S(i) \neq n + 1 = S(n)$.

2.2 De cykel: C_n

De cykel van lengte $n > 2$, C_n , is een graaf met n punten en n kanten, zodat voor elk punt x , $d(x) = 2$. Dit betekent dat we de punten zo kunnen nummeren van $1, \dots, n$ dat $E_{C_n} = \{\langle i, i + 1 \rangle : i \in \{1, \dots, n - 1\}\} \cup \{\langle 1, n \rangle\}$. Hieronder is een afbeelding gegeven van de C_6 :



Om aan te tonen dat de C_n anti-magisch is voor alle n , zullen we opnieuw onderscheid maken tussen even en oneven n :

- Als n oneven is, dan is de volgende afbeelding een anti-magische labeling:

$$W(\langle i, i+1 \rangle) = i \quad (i \in \{1, \dots, n-1\}).$$

Bewijs: We zien dat $S(1) = n+1$. Voor alle $i \in \{2, \dots, n\}$ geldt $S(i) = (i-1) + i = 2i-1$. Duidelijk is dat geldt $S(i) = 2i-1 \neq 2j-1 = S(j)$ als $i \neq j$, $i, j \in \{2, \dots, n\}$. Verder is $S(i)$ oneven, dus $S(i) \neq n+1 = S(1)$ voor $i \in \{2, \dots, n\}$.

- Als n even is, dan is de volgende afbeelding een anti-magische labeling:

$$W(\langle i, i+1 \rangle) = i \quad (i \in \{1, \dots, n-2\})$$

$$W(\langle n-1, n \rangle) = n$$

$$W(\langle 1, n \rangle) = n-1$$

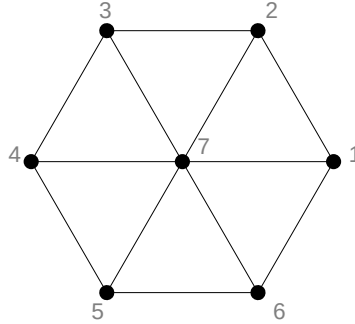
Bewijs: We zien dat $S(1) = n$, $S(n-1) = n + (n-2) \neq n = S(1)$. Voor $i \in \{2, \dots, n-2\} \cup \{n\}$ geldt $S(i) = (i-1) + i = 2i-1$. Omdat $S(1)$ en $S(n-1)$ beide even zijn, terwijl alle onderling verschillende $S(i) : i \in \{2, \dots, n-2\} \cup \{n\}$ oneven zijn, is W een anti-magische labeling.

2.3 Het wiel: W_n

Het wiel met $n > 2$ spaken, W_n , is een graaf met $n+1$ punten en $2n$ kanten, zodat voor één punt x geldt $d(x) = n$ en voor alle andere punten y : $d(y) = 3$. We kunnen de punten van de graaf zo nummeren dat

$$E_{W_n} = \{\langle i, i+1 \rangle : i \in \{1, \dots, n-1\}\} \cup \{\langle i, n+1 \rangle : i \in \{1, \dots, n-1\}\}.$$

Hieronder zien we een afbeelding van de W_6 :



We willen aantonen dat de W_n anti-magisch is voor alle n . Hiervoor bekijken we twee gevallen: het geval dat n een drievoud is en het geval dat n geen drievoud is.

- Als n geen drievoud is, dan is de volgende afbeelding een anti-magische labeling:

$$W(\langle i, n+1 \rangle) = 2i-1 \quad (i \in \{1, \dots, n\})$$

$$W(\langle i, i+1 \rangle) = 2i \quad (i \in \{1, \dots, n-1\})$$

$$W(\langle 1, n \rangle) = 2n$$

Bewijs: Voor het middelste punt $n+1$ geldt $S(n+1) = \sum_{i=1}^n 2i-1 = n^2$. Daarnaast zien we dat $S(1) = 2n+1+2 = 2n+3 (\neq n^2 \text{ voor } n \neq 3)$. Voor de overige punten $S(i) = 2(i-1) + 2i + 2i-1 = 6i-3$ ($i \in \{2, \dots, n\}$); wat betekent dat voor deze punten geldt $(i \neq j) \rightarrow S(i) \neq S(j)$. Omdat $6i-3$ een drievoud is, terwijl n dat niet is, geldt ook $(i \neq 1) \rightarrow [S(i) \neq 2n+3 = S(1)]$ en $(i \neq n+1) \rightarrow [S(i) \neq n^2 = S(n+1)]$.

- Als n een drievoud is, dan is de volgende afbeelding een anti-magische labeling:

$$W(\langle i, n+1 \rangle) = 2i - 1 \quad (i \in \{1, \dots, n-1\})$$

$$W(\langle i, i+1 \rangle) = 2i \quad (i \in \{1, \dots, n-1\})$$

$$W(\langle 1, n \rangle) = 2n - 1$$

$$W(\langle n, n+1 \rangle) = 2n$$

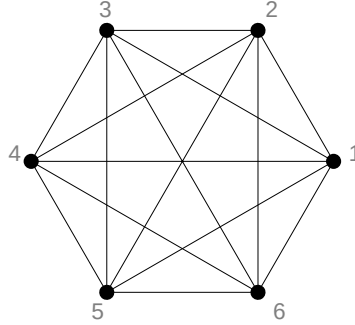
Bewijs: In dit geval zien we dat voor het middelste punt $n+1$ geldt

$$S(n+1) = 2n + \sum_{i=1}^{n-1} 2i - 1 = 2n + (n-1)^2 = n^2 + 1. \text{ Ook zien we dat}$$

$S(1) = 2n - 1 + 1 + 2 = 2n + 2 \neq n^2 + 1$ voor alle $n \in \mathbb{N}$. Omdat n een drievoud is, zijn zowel $S(1)$ als $S(n+1)$ geen drievoud. Aan de overige punten is niets veranderd ten opzichte van het vorige geval, dus $S(i) = 6i - 3$ ($i \in \{2, \dots, n\}$); dit zijn dus (onderling verschillende) drievouden, waardoor ze ongelijk zijn aan zowel $S(1)$ als $S(n+1)$.

2.4 De volledige graaf: K_n

De volledige graaf op n punten is een graaf waarin alle punten onderling verbonden zijn. Nummeren we de punten van de graaf van 1 tot n , dan geldt dat $E_{K_n} = \{\langle i, j \rangle : i < j \wedge i, j \in \{1, \dots, n\}\}$. Hieronder zien we een afbeelding van de K_6 :



Voordat we een anti-magische labeling geven voor deze familie van grafen, eerst enkele definities:

Definitie: Zij G een graaf met p punten en $N : V_G \rightarrow \{1, \dots, p\}$ een nummering van de punten. Dan definiëren we de verzameling van geordende paren punten:

$$E_G^* = \{(a, b) \mid a, b \in V_G \mid \langle a, b \rangle \in E_G \wedge N(a) < N(b)\}.$$

Definitie: Zij G een graaf. Bij elke nummering van de punten $N : V_G \rightarrow \{1, \dots, p\}$ maken we twee operatoren $\min^N$ en $\max^N$, als volgt:

$$\begin{aligned} \min^N(a, b) &= a && \text{als } N(a) \leq N(b) \\ &= b && \text{als } N(b) < N(a) \\ \max^N(a, b) &= a && \text{als } N(a) \geq N(b) \\ &= b && \text{als } N(b) > N(a) \end{aligned}$$

Definitie: Zij G een graaf en $N : V_G \rightarrow \{1, \dots, p\}$ een nummering van de punten.

We definiëren een relatie $<_{lex}^* \subseteq E_G^* \times E_G^*$ door

$$(a, b) <_{lex}^* (c, d) \Leftrightarrow (N(a) < N(c)) \vee (N(a) = N(c) \wedge N(b) < N(d)).$$

Op een natuurlijke manier verkrijgen we nu de relatie die we gaan gebruiken in ons bewijs:

Definitie: Zij G een graaf en $N : V_G \rightarrow \{1, \dots, p\}$ een nummering van de punten.

Dan definiëren we een relatie $<_{lex} \subseteq E_G \times E_G$ door:

$$\langle a, b \rangle <_{lex} \langle c, d \rangle \Leftrightarrow (Min^N(a, b), Max^N(a, b)) <_{lex}^* (Min^N(c, d), Max^N(c, d)).$$

Zij nu $G = K_n$, $N : V_{K_n} \rightarrow \{1, \dots, n\}$ een nummering van de punten en $<_{lex}$ de zojuist gedefinieerde tweestemmige relatie op E_{K_n} . Dan kunnen we een eindige rij $a_1, a_2, \dots, a_{n(n-1)}$ ($a_i \in E_{K_n}$) maken, zodat voor elke $i, j \in \{1, \dots, n(n-1)\}$: $i < j \rightarrow (a_i <_{lex} a_j)$. Dan is de afbeelding $W(a_i) = i$ een anti-magische labeling.

Bewijs: We laten zien dat voor twee punten a, b geldt: als $N(a) < N(b)$, dan $S(a) < S(b)$. Zij a en b punten van de graaf. Dan $S(a) = \sum_{x \in B(a)} W(\langle a, x \rangle) = W(\langle a, b \rangle) + \sum_{x \in B(a) \setminus \{b\}} W(\langle a, x \rangle)$, terwijl $S(b) = \sum_{x \in B(b)} W(\langle b, x \rangle) = W(\langle a, b \rangle) + \sum_{x \in B(b) \setminus \{a\}} W(\langle b, x \rangle)$. In de K_n geldt dat $B(b) \setminus \{a\} = B(a) \setminus \{b\}$ (we noemen deze verzameling vanaf nu P), dus moeten we aantonen dat $\sum_{x \in P} W(\langle a, x \rangle) < \sum_{x \in P} W(\langle b, x \rangle)$. Voor elke $x \in P$ geldt $\langle a, x \rangle <_{lex} \langle b, x \rangle$ (want $N(x) = N(x)$ en $N(a) < N(b)$), dus $W(\langle a, x \rangle) < W(\langle b, x \rangle)$. Dit betekent dat $S(a) < S(b)$.

2.5 Dichte grafen

Het feit dat de K_n anti-magisch is voor elke n , hadden we ook anders kunnen zien.

Het is namelijk zo dat voor elke graaf G met p punten en q kanten:

$\exists a \in V_G [d(a) = p - 1] \rightarrow (G \text{ is anti-magisch}).$

Dat wil zeggen dat elke graaf G met een punt van graad $p - 1$ anti-magisch is.

Bewijs: Zij G een graaf en a een punt van G met graad $p - 1$.

Zij $E^a = \{\langle a, x \rangle : x \in B(a)\}$, $W^* : E_G \setminus E^a \rightarrow \{1, \dots, q - p + 1\}$ een willekeurige injectieve afbeelding.

Definieer nu voor alle $x \in V_G \setminus \{a\}$: $S^*(x) = \sum_{y \in B(x) \setminus \{a\}} W^*(\langle x, y \rangle)$. Construeer nu een rij $z_1, z_2, \dots, z_{p-1}$ ($z_i \in V_G \setminus \{a\}$, zodat $i < j \rightarrow S^*(z_i) \leq S^*(z_j)$) (we ordenen de punten in $V_G \setminus \{a\}$ volgens hun labelsom). Dan maken we de anti-magische labeling:

$$\begin{aligned} W(\langle x, y \rangle) &= W^*(\langle x, y \rangle) & \text{als } x, y \in V_G \setminus \{a\} \\ W(\langle a, x \rangle) &= q - p + 1 + i & \text{als } x = z_i \end{aligned}$$

We zien nu dat $S(a) = \sum_{j=q-p+2}^q j$; de maximaal mogelijke waarde in deze graaf, dus: $\forall x \neq a [S(x) < S(a)]$. Voor de overige punten $\{z_1, \dots, z_{p-1}\}$ geldt:

$i < j \rightarrow S(z_i) < S(z_j)$. We merken eerst op dat voor alle $i \in \{1, \dots, p-1\}$:

$i < j \rightarrow S^*(z_i) \leq S^*(z_j)$ en constateren dat $S(z_i) = S^*(z_i) + W(\langle z_i, a \rangle) = S^*(z_i) + q - p + 1 + i$. Nu kunnen we concluderen dat

$i < j \rightarrow [S(z_i) = (S^*(z_i) + q - p + 1 + i) < (S^*(z_j) + q - p + 1 + j) = S(z_j)]$ en dus dat G anti-magisch is.

In [3] is met behulp van probabilistische lemma's aangetoond dat ook alle grafen G met een punt van graad $p - 2$ anti-magisch zijn.

2.6 Verdere bekende resultaten

Voor meer bekende resultaten over anti-magische labelingen verwijs ik naar [5] en [6]. De technieken die in deze artikelen gebruikt worden, vereisen veel uitleg en worden in deze scriptie verder niet meer genoemd.

3 Het genereren van bomen

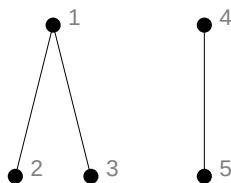
Om een goed beeld te krijgen van de aannemelijkheid van het vermoeden “Alle grafen, behalve K_2 zijn antimagisch”, ben ik gaan tellen hoeveel mogelijke anti-magische labelingen een bepaalde graaf heeft. Op deze manier ben ik op zoek gegaan naar ‘moeilijke’ grafen om te kijken welke eigenschappen deze hebben, die minder ‘moeilijke’ grafen niet hebben. Het begrip ‘moeilijk’ heb ik hierbij geïnterpreteerd als volgt:

Zij voor een graaf G het getal $A(G)$ het aantal anti-magische labelingen van G , dan noemen we G *moeilijker* dan H , als $\frac{A(G)}{(\#E_G)!} < \frac{A(H)}{(\#E_H)!}$. Anders gezegd, betekent dit dat de kans dat je bij een willekeurige labeling van G een anti-magische labeling hebt, kleiner is dan bij H .

Om erachter te komen of een gegeven graaf moeilijker is dan een andere, dienen dus alle mogelijke labelingen gecontroleerd te worden op hun ‘anti-magisch’ zijn. Vervelend hierbij is dat het aantal mogelijke labelingen gelijk is aan $(\#E_G)!$; een aantal dat erg snel stijgt als de graaf meer kanten bezit. Een ander probleem is het systematisch afgaan van de grafen, zonder daarbij isomorfieën van al eerder behandelde grafen opnieuw te testen. Om deze twee problemen zo klein mogelijk te maken, heb ik besloten om me vooral te richten op het grote deelvermoeden “Alle bomen, behalve K_2 , zijn anti-magisch”. Bomen hebben namelijk het voordeel dat ze weinig kanten hebben (in vergelijking tot andere grafen met hetzelfde aantal punten). Daarnaast staat in [2] (bladzijde 25) een algoritme beschreven waarmee het mogelijk is, bij een gegeven aantal punten, alle vrije bomen te genereren. In de volgende deelparagrafen zal ik beschrijven hoe dit algoritme in zijn werk gaat.

3.1 Alle gerichte bossen genereren

Het algoritme dat we uiteindelijk willen hebben, geeft uit elke isomorfieklasse één boom terug. Dit is gebaseerd op een ander algoritme, dat uit elke isomorfieklasse één gericht bos genereert. In een gericht bos heeft elk punt maximaal één ouder; dit is de buur die direct boven het punt ligt. In de afbeelding hieronder is 1 dus de ouder van 2 en 3, terwijl 4 de ouder is van 5. 1 en 5 hebben geen ouder, dus geven we ze de denkbeeldige ouder 0.



Een gericht bos met n punten, wordt gerepresenteerd door een rijtje $[p_1, p_2, \dots, p_n]$, waarbij p_i betekent dat het punt p_i ouder is van het punt i . Het bos hierboven wordt dus gerepresenteerd als $[0, 1, 1, 0, 4]$.

We zullen het algoritme voor de bossen (zie ook 3.2) wat nader bekijken, aangezien we dit ook nodig hebben voor het genereren van alle bomen. Het idee is dat we een boom invoeren, hieraan systematisch één, of meerdere takken veranderen en het nieuwe bos teruggeven. Dit gebeurt zo, dat we elk bos maar één keer tegenkomen. Tijdens het proces bewaren we steeds de bossen die we al gevonden hebben, zodat we aan het eind van het algoritme een rijtje hebben met alle gerichte bossen van n punten.

We zullen het algoritme beginnen met het bos $[0, 1, 2, \dots, n-1]$ en eindigen als we het bos $[0, 0, \dots, 0]$ gevonden hebben. Als we tijdens het proces een bos $[p_1, p_2, \dots, p_n]$ gevonden hebben, kijken we in de volgende stap allereerst naar p_n . Als deze ongelijk is aan 0, dan wordt hij gelijk aan p_{p_n} . In andere woorden: als n geen boomtop is, dan wordt zijn nieuwe ouder, het punt dat eerst zijn grootouder (de ouder van zijn ouder) was. Nu hebben we een nieuw bos gevonden (in de eerste stap vinden we dus $[0, 1, 2, \dots, n-2, n-2]$).

Wanneer p_n echter gelijk is aan 0, dan zoeken we naar de grootste k , zodat $p_k \neq 0$. Als deze k niet bestaat, dan zijn we in het geval $[0, 0, \dots, 0]$, dus zijn we klaar. Bestaat deze k wel, dan definiëren we j als p_k en d als $k - j$. Daarna wordt de nieuwe ouder van k het punt dat eerst zijn grootouder was en kijken we voor de punten $i \in \{k+1, \dots, n\}$ wat de nieuwe ouders moeten worden. Hierbij onderscheiden we twee gevallen: als $p_{i-d} = p_j$, dan wordt de nieuwe ouder van i de ouder van j (oftewel de nieuwe ouder van k). Is echter $p_{i-d} \neq p_j$, dan wordt de nieuwe ouder van i , het punt $p_{i-d} + d$ (de top van de boom waarin het punt k hangt).

Ter illustratie zullen we voor $n = 4$ kijken welke bossen we tegenkomen:

- We beginnen met $[0, 1, 2, 3]$;
- Omdat het punt 4 geen boomtop is, wordt zijn nieuwe ouder de ouder van 3, dus krijgen we $[0, 1, 2, 2]$;
- Op dezelfde manier verkrijgen we nu $[0, 1, 2, 1]$ en $[0, 1, 2, 0]$;
- Nu is $n = 4$ een boomtop (zijn ouder is 0), dus kijken we naar het punt $k = 3$. De ouder van dit punt wordt nu zijn grootouder, dus werken we met het rijtje $[0, 1, 1, 0]$. De stap is echter nog niet afgelopen, dus we definiëren nu $j = 2$ en $d = 3 - 2 = 1$. Voor het punt $k + 1 = 4$ kijken we of $p_{k+1-d} = p_{4-1} = p_3$ gelijk is aan $p_j = p_2$. Dit is het geval, dus wordt de nieuwe ouder van $k + 1 = 4$ het punt $j = 2$. Het nieuwe gevonden rijtje is dus $[0, 1, 1, 1]$;
- Op dezelfde manier als eerder vinden we nu het rijtje $[0, 1, 1, 0]$;
- Nu is $n = 4$ opnieuw een boomtop, dus kijken we naar het punt $k = 3$. Zijn ouder wordt nu zijn grootouder, dus hebben we het tijdelijke rijtje $[0, 1, 0, 0]$. Daarna definiëren we $j = 1$ en $d = 3 - 1 = 2$. Voor het punt $k + 1 = 4$ geldt nu dat $p_{k+1-d} = p_{4-2} = p_2 = 1$ niet gelijk is aan $p_j = p_1 = 0$. Dus wordt de nieuwe ouder van 4 het punt $p_{k+1-d} + d = p_{4-2} + 2 = 1 + 2 = 3$. Het gevonden rijtje is dus $[0, 1, 0, 3]$;
- Omdat 4 nu weer geen boomtop is, wordt zijn nieuwe ouder de ouder van 3, oftewel 0. Zo vinden we $[0, 1, 0, 0]$;
- Tot slot komen we in het geval dat $k = 2$ de grootste niet-top is. We verkrijgen nu het rijtje $[0, 0, 0, 0]$. Voor de punten 3 en 4 geldt nu dat ze de ouder van 1 als nieuwe ouder krijgen ($p_i = p_j$ voor alle i, j , dus is aan de voorwaarde $p_{i-d} = p_j$ automatisch voldaan), dus behouden we het rijtje $[0, 0, 0, 0]$.

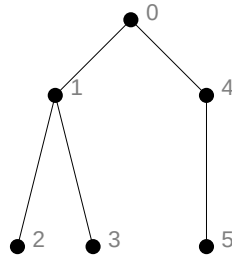
3.2 Implementatie in Magma

Het in de vorige paragraaf beschreven algoritme heb ik geprogrammeerd met behulp van het computeralgebrapakket Magma. De code ziet er als volgt uit:

```
GenereerBossen:=function(n);
  p:=[i-1: i in [1..n]];
  Bossen:=[];
  while p ne [] do
    Append(~Bossen,p);
    if p[n] gt 0 then
      p[n]:=p[p[n]];
    else
      k:=n-1;
      while k gt 0 and p[k] eq 0 do
        k:=k-1;
      end while;
      if k eq 0 then
        p:=[];
      else
        j:=p[k];
        d:=k-j;
        repeat
          if p[k-d] eq p[j] then
            p[k]:=p[j];
          else
            p[k]:=p[k-d]+d;
          end if;
          k:=k+1;
        until k eq n+1;
      end if;
    end if;
  end while;
  return Bossen;
end function;
```

3.3 Het genereren van alle vrije bomen

In de vorige paragrafen hebben we gezien hoe we alle gerichte bossen met n punten kunnen genereren. Hierbij hebben we een gericht bos weergegeven als een rijtje $[p_1, p_2, \dots, p_n]$ ($p_i \in \{0, \dots, n\}$). We kunnen dit rijtje echter ook opvatten als een boom met $n + 1$ punten, door het punt 0 toe te voegen. Het rijtje $[0, 1, 1, 0, 4]$, wat we in 3.1 als voorbeeld hebben gezien, stelt dan de volgende boom voor:



Op deze manier kunnen we bij elk gericht bos met n punten een unieke boom met $n + 1$ punten vinden. Echter, het is niet zo dat we bij elke boom een uniek gericht bos kunnen vinden. Kijk bijvoorbeeld naar de rijtjes $[0, 0, 0]$ en $[0, 1, 1]$. Hoewel deze rijtjes verschillende bossen opleveren, zijn ze beide representant van dezelfde boom. Dit zien we in het volgende plaatje:



Op het eerste oog hebben we nu twee verschillende bomen, maar het is niet moeilijk in te zien dat deze bomen isomorf zijn (punt 0 correspondeert met punt 1 en vice versa). Omdat het anti-magisch zijn van een boom niet afhangt van de nummering van de punten, willen we in ons proces niet twee isomorfe bomen hoeven te testen. Daarom moeten we het algoritme van de bossen iets aanpassen.

Allereerst gaan we alle bomen genereren die één centroïde hebben ($\#C_B = 1$). We zorgen dat het punt 0 de enige centroïde is, door alle bossen te genereren (met het algoritme uit 3.1), waarbij elke boom in het bos hoogstens $\lfloor \frac{n}{2} \rfloor$ punten heeft. Om dit te doen, nemen we het algoritme uit 3.1 als basis en veranderen er enkele kleine dingen in. Hoe dit algoritme aangepast moet worden en waarom, is te vinden in [?]

Om te beginnen veranderen we de beginstand: hier bestaat het bos namelijk uit één boom met n punten. We definiëren m als $\lfloor \frac{n}{2} \rfloor$ en zetten $p_{m+1} = 0$ en als n oneven ook $p_{2m+1} = 0$. Nu hebben we dus een bos met twee bomen van m punten, plus een eventuele derde boom van één punt.

Wanneer het laatste punt n geen boomtop is (dus $p_n \neq 0$), doen we hetzelfde als eerder: zijn grootouder wordt zijn nieuwe ouder. Wanneer n wel een boomtop is, dan gaan we zoals eerder op zoek naar de grootste k die geen boomtop is. Als er niet zo een k is, dan stoppen we het algoritme (we eindigen dus ook in dit algoritme met het rijtje $[0, 0, \dots, 0]$). Is er wel zo een k , dan wordt deze kind van zijn grootouder en maken we op dezelfde manieren als in 3.1 de variabelen $j = p_k$ en $d = k - j$ aan. Daarnaast maken we echter ook een nieuwe variabele i aan middels het volgende proces: initialiseer $i = j$; zolang $p_i \neq 0 : i = p_i$. Op deze manier wordt i de top van de boom waar k en j in zitten.

Nu moeten we voor de punten $a \in \{k + 1, \dots, n\}$ kijken wat de nieuwe ouders worden. We willen echter niet teveel punten aan dezelfde boom hangen, dus als $a = i + m$ en $i < j$, dan zetten we $j = i$ en $d = m$ (we schuiven het punt als het ware naar de volgende boom). Daarna gaan we verder op dezelfde manier als bij het oude algoritme: als $p_{a-d} = p_j$, dan wordt de nieuwe ouder van a de ouder van j (oftewel de nieuwe ouder van k). Is echter $p_{a-d} \neq p_j$, dan wordt de nieuwe ouder van a , het punt $p_{a-d} + d$.

Daarna moeten we nog alle bomen genereren waarvoor ($\#C_B = 2$). Als $\#V_B$ oneven is, dan bestaan er niet zulke bomen. Als $\#V_B$ echter even is, dan missen we nog wel enkele bomen. Daarom gaan we op zoek naar alle gerichte bossen met $m = \frac{n}{2}$ punten; noem deze verzameling $F^{(n)}$. Bekijken we $F^{(n)}$ als de verzameling van de rijtjes die corresponderen bij de bossen, dan kunnen we een ordening $\leq^* \subset F^{(n)} \times F^{(n)}$ maken door, voor $p = [p_1, \dots, p_m]$ en $q = [q_1, \dots, q_m] \in F^{(n)}$:

$$p \leq^* q \quad \text{als} \quad \exists i [p_i < q_i \wedge \forall j < i [p_i = q_i]].$$

Kijken we nu naar alle paren p, q van bossen met m punten, zodat $p \leq^* q$, dan definiëren we een nieuwe boom (aan de hand van zijn corresponderende rijtje) b door:

$$\begin{aligned} b_1 &= 0 \\ b_{j+1} &= p_j + 1 & (1 \leq j \leq m) \\ b_{m+j+1} &= q_j + m + 1 \quad \text{als} \quad q_j > 0 & (1 \leq j \leq m). \end{aligned}$$

Dit levert ons alle bomen met $n + 1$ punten.

3.4 Implementatie in Magma

Het in de vorige paragraaf beschreven algoritme ziet er in Magma als volgt uit:

```
GenereerBomen:=function(n);

    p:=[i-1: i in [1..n]];
    m:=Floor(n/2);
    p[m+1]:=0;
    if IsOdd(n) then
        p[2*m+1]:=0;
    end if;
    Bomen:=[];
    while p ne [] do
        Append(~Bomen,p);
        if p[n] gt 0 then
            p[n]:=p[p[n]];
        else
            k:=n-1;
            while k gt 0 and p[k] eq 0 do
                k:=k-1;
            end while;
            if k eq 0 then
                p:=[];
            else
                j:=p[k];
                d:=k-j;
                i:=j;
                while p[i] ne 0 do
                    i:=p[i];
                end while;
                repeat
                    if k eq i+m and i lt j then
                        j:=i;
                        d:=m;
                    end if;
                    if p[k-d] eq p[j] then
                        p[k]:=p[j];
                    else
                        p[k]:=p[k-d]+d;
                    end if;
                    k:=k+1;
                until k eq n+1;
            end if;
        end if;
    end while;
```

```

if IsOdd(n) then
  Bossen:=GenereerBossen(m);
  aantal:=#Bossen;
  for a:=1 to aantal do
    for b:=a to aantal do
      if Bossen[a] ge Bossen[b] then
        s:=Bossen[a]; t:=Bossen[b];
      else
        s:=Bossen[b]; t:=Bossen[a];
      end if;
    end for;
  m:=Floor(n/2);
  p:= [0: i in [1..n]];
  for l:=1 to m do
    p[l+1]:=s[l]+1;
    if t[l] gt 0 then
      p[m+l+1]:=t[l]+m+1;
    end if;
  end for;
  if not p in Bomen then
    Append(~Bomen,p);
  end if;
end for;
end if;
return Bomen;
end function;

```

4 Het vinden van anti-magische labelingen

In het vorige hoofdstuk hebben we gezien hoe we alle bomen op een relatief snelle manier kunnen genereren, zodat we hiermee kunnen experimenteren. In dit hoofdstuk zullen we kijken naar de algoritmes die ik in Magma heb geschreven om bij een gegeven boom alle anti-magische labelingen te geven. Het doel hiervan is om te kijken of er bepaalde bomen zijn die slechts op enkele specifieke manieren te labelen zijn; als we deze vinden, dan kunnen we eigenschappen van deze bomen bestuderen en op deze manier wellicht een constructie verzinnen voor een boom (of graaf) die niet anti-magisch te labelen is. Zijn er daarentegen voor elke boom erg veel anti-magische labelingen, dan lijkt de juistheid van het vermoeden ‘waarschijnlijker’. De resultaten van de experimenten zijn te lezen in het volgende hoofdstuk; eerst zullen we kijken naar het gebruikte algoritme (en zijn hulpalgoritmes).

4.1 De raamfunctie: ‘Testallebomen’

De functie ‘Testallebomen’ is een functie die in feite alleen voor de overzichtelijkheid gemaakt is. In deze functie wordt een rijtje met bomen (die weergegeven worden als rijtjes $[p_1, p_2, \dots, p_n]$ als in het vorige hoofdstuk) ingevoerd en er worden twee rijtjes teruggegeven:

1. Het rijtje ‘Kansen’ waarin op plek i staat welk percentage van alle mogelijke labelingen van boom i anti-magisch is.
2. Het rijtje ‘Labelingen’ waarin op plek i een verzameling staat met daarin alle anti-magische labelingen van boom i .

Om deze twee rijtjes te vinden, worden de hulpfunctie ‘VertaalNaarMatrix’ en ‘antimagischboom’ aangeroepen. Hoe deze in elkaar steken is te lezen in paragraaf 4.2 respectievelijk 4.5.

4.1.1 Implementatie in Magma

```
Testallebomen:=function(bomen);

    i:=0;
    Kansen:=[];
    Labelingen:=[];
    for k:=1 to #bomen do
        i:=i+1;
        rijtje:=bomen[i];
        G:=VertaalNaarMatrix(rijtje);
        H:=antimagischboom(G);
        Append(~Kansen,#H/Factorial(#rijtje));
        Append(~Labelingen,H);
    end for;
    return Kansen,Labelingen;
end function;
```

4.2 Hulpfunctie 1: ‘VertaalNaarMatrix’

Om de kanten van een boom weer te geven, is het makkelijker om de boom naar een ‘matrix’ (of beter: een rijtje van rijtjes) te vertalen. In feite gebruiken we een beneden-driehoeksmatrix. De functie ‘VertaalNaarMatrix’ heeft als invoer een rijtje p , van lengte n , dat een boom met $n + 1$ punten voorstelt, en als uitvoer een matrix M die aan de volgende eigenschappen voldoet:

1. M heeft $n + 1$ rijen, waarbij rij i lengte $i - 1$ heeft;
2. Als $\langle i, j \rangle \in E_B \wedge i > j$, dan $M(i + 1, j + 1) = -1$;
3. Als $\langle i, j \rangle \notin E_B \wedge i > j$, dan $M(i + 1, j + 1) = 0$;
4. Als $i \leq j$, dan is $M(i, j)$ niet gedefinieerd.

Als we nu een bepaalde waarde w aan een kant willen geven, kunnen we dat doen door in deze matrix de waarde -1 te vervangen door w . Dit zal gebeuren in de functie ‘antimagischboom’ (4.5).

4.2.1 Implementatie in Magma

```
VertaalNaarMatrix:=function(p);  
  
  M:=[];  
  for j:=1 to #p+1 do  
    Append(~M,[0: i in [1..j-1]]);  
  end for;  
  for i:=2 to #p+1 do  
    M[i,p[i-1]+1]:=-1;  
  end for;  
  return M;  
end function;
```

4.3 Hulpfunctie 2: ‘Puntsom’

De functie ‘Puntsom’ is een hulpfunctie die gebruikt wordt in de functie ‘antimagischboom’. De invoer van de functie is een matrix (soortgelijk als de uitvoer van de functie ‘VertaalNaarMatrix’), waarbij op elke kant (de oorspronkelijke -1 ’en) een waarde (uit $\{1, \dots, n\}$) is ingevuld, en de uitvoer is een rijtje ‘sommen’ van lengte $n + 1$ (het aantal punten van de boom) met op plaats i de som van de waarden van alle kanten waarin het punt i optreedt. Om deze waarden op te tellen, moeten we kijken naar de waarden in de matrix op de plaatsen $(i + 1, x)$ en $(x, i + 1)$, waarbij $1 \leq x \leq i$. Deze tellen we achtereenvolgens op voor elke i .

Aan het einde wordt het rijtje ‘sommen’ omgezet naar de verzameling ‘somverz’. Dit gebeurt omdat we dan enkel het aantal elementen van de verzameling te hoeven tellen om te concluderen dat alle sommen verschillend zijn.

4.3.1 Implementatie in Magma

```
puntsom:=function(M);  
  
    n:=#M;  
    sommen:=[];  
    for i:=1 to n do  
        som:=0;  
        for j:=1 to i-1 do  
            som:=som+M[i,j];  
        end for;  
        for j:=i+1 to n do  
            som:=som+M[j,i];  
        end for;  
        Append(~sommen,som);  
    end for;  
    somverz:=SequenceToSet(sommen);  
    return sommen,somverz;  
end function;
```

4.4 Hulpfunctie 3: ‘VolgendeRijetje’

Om alle mogelijke labelingen op een logische volgorde te testen, is de functie ‘VolgendeRijetje’ ontwikkeld. In de functie ‘antimagischboom’ worden aan de kanten in de eerder gecreëerde matrix een waarde toegewezen; deze waarden worden afgelezen uit een rijtje $a = [a_0, a_1, \dots, a_n]$. Als deze labeling eenmaal getest is, moet er een nieuwe labeling gegeven worden. We lopen de rijtjes in lexicografische volgorde af, dus de functie ‘VolgendeRijetje’ geeft bij een gegeven rijtje a het volgende rijtje terug in de lexicografische ordening op de elementen $0, \dots, n$.

De functie test eerst of we te maken hebben met het rijtje $[n, n-1, \dots, 1]$; als dit zo is, stoppen we het algoritme. Als dit echter niet zo is, dan gaan we op zoek naar de grootste k , zodat $a_k < a_{k+1}$. Dan zoeken we de kleinste a_i voor $i \in \{k+1, \dots, n\}$ en verwisselen deze met a_k . Daarna zetten we het gedeelte $[a_{k+1}, \dots, a_n]$ achterstevoren. Dit levert het nieuwe rijtje.

4.4.1 Implementatie in Magma

```
Volgenderijjtje:=function(a);

q:=#a;
if a eq [q..1 by -1] then
    return [0: i in [1..q]];
else
    H:=[a[q]];
    k:=q-1;
    while a[k] gt a[k+1] do
        Append(~H,a[k]);
        k:=k-1;
    end while;
    j:=Min([a[i]: i in [1..q] | a[i] gt a[k] and i gt k]);
    for i:=k+1 to q do
        if a[i] eq j then
            l:=i;
            break i;
        end if;
    end for;
    z:=a[k];
    a[k]:=j;
    a[l]:=z;
    b:=a;
    for i:=k+1 to q do
        a[i]:=b[q+k+1-i];
    end for;
end if;
return a;
end function;
```

4.5 De hoofdfunctie: ‘Antimagischboom’

Nu de hulpfuncties allemaal gedefinieerd zijn, is de werking van de functie ‘antimagischboom’ eenvoudig uit te leggen. De invoer van de functie is een matrix M , zoals deze uit de functie ‘VertaalNaarMatrix’ komt. We starten met het rijtje $a = [1, \dots, n]$. We gaan nu in de matrix M de eerste -1 zoeken en vervangen deze door de waarde a_1 . Daarna zoeken we verder naar de volgende -1 en vervangen deze door a_2 . Zo gaan we door tot we alle waarden uit a in de matrix hebben geplaatst. Op dat moment voeren we de functie puntsom uit op onze matrix, die ons een rijtje en een verzameling met sommen teruggeeft. Als het aantal elementen van de verzameling gelijk is aan het aantal punten, dan gaf het rijtje a een anti-magische labeling, dus stoppen we het in de verzameling G . Zo niet, dan doen we niets met het rijtje.

Ten slotte laten we de functie ‘VolgendeRijjtje’ het volgende rijtje bepalen dat we gaan testen. Deze stappen voeren we uit tot we alle rijtjes hebben getest. De functie geeft dan de verzameling G uit met alle mogelijke anti-magische labelingen.

4.5.1 Implementatie in Magma

```
antimagischboom:=function(M);

q:=#M-1;
G:={};
a:=[1..q];
N:=M;
for aantal:=1 to Factorial(q) do
    n:=1;
    for i:=1 to #N do
        for j:=i+1 to #N do
            if N[j,i] eq -1 then
                N[j,i]:=a[n];
                n:=n+1;
            end if;
        end for;
    end for;
    som,verz:=puntsom(N);
    if #verz eq #N then
        G:=G join {a};
    end if;
    a:=Volgenderijjtje(a);
    N:=M;
end for;
return G;
end function;
```

5 Resultaten, verbeterpunten en verdere zoekrichtingen

Nadat ik alle functies uit de vorige twee hoofdstukken geïmplementeerd heb, heb ik Magma twee weken lang aan het werk gezet. De resultaten die dit opgeleverd heeft, zal ik in dit hoofdstuk bespreken. Daarnaast zal ik hierbij enkele voorzichtige conclusies trekken en verbeterpunten aangeven die voor meer, of betere resultaten gezorgd zouden hebben. Tot slot zal ik enkele ideeën geven voor verdere zoekpogingen in deze richting, om zo wellicht dichter bij het bevestigen, danwel ontkrachten van het vermoeden te komen.

5.1 Resultaten

Om te beginnen heb ik de functie ‘GenereerBomen’ twee weken laten lopen voor verschillende waarden n . Resultaat hiervan is dat ik alle vrije bomen met n punten ($n \in \{1, \dots, 20\}$) in twintig verschillende bestanden heb opgeslagen. Dit is op zichzelf een mooie database voor verschillende soorten experimenten waarin bomen een rol spelen. Helaas is 20 geen heel groot aantal, waardoor ik weinig gegevens gekregen heb op ‘grote’ bomen.

Tegelijkertijd heb ik in Magma de functie ‘TestAlleBomen’ laten draaien, waarbij ik achtereenvolgens de aangemaakte bestanden met bomen heb ingeladen en alle bomen uit één bestand in een rijtje als invoer heb meegegeven. De resultaten van deze berekening heb ik weer weggeschreven in een bestand waarin achtereenvolgens de rijtjes ‘Kansen’ en ‘Labelingen’ worden geprint. Het aantal benodigde berekeningen loopt echter zo schrikbarend snel op (sneller dan $n!$), dat hieruit enkel de resultaten voor bomen met 9 of minder punten gereed waren binnen twee weken. Wellicht zijn dit te weinig resultaten om conclusies aan te verbinden, maar de patronen die ik gevonden heb, zal ik toch opschrijven.

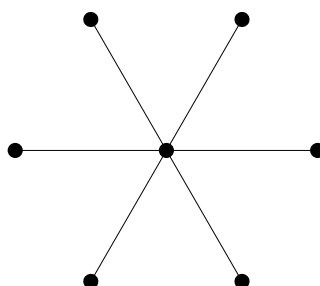
- * Allereerst blijkt, niet verrassend, dat alle bomen die ik getest heb ook daadwerkelijk anti-magisch te labelen zijn. Dit betekent dat er geen tegenvoorbeelden te vinden zijn met minder dan 9 punten.
- * Opvallender is het aantal anti-magische labelingen dat bij elke boom te vinden is. In de volgende tabel staat bij elke n (het aantal punten) het aantal anti-magische labelingen van de boom met n punten waarbij dit aantal het kleinste is, en de kleinste kans om een anti-magische labeling te vinden.

n	Aantal labelingen	Kans
3	2	1
4	2	$\frac{1}{2}$
5	6	$\frac{1}{3}$
6	16	$\frac{2}{5}$
7	92	$\frac{15}{23}$
8	352	$\frac{180}{22}$
9	1910	$\frac{315}{191}$
		4032

We zien in de tabel dat het aantal anti-magische labelingen erg snel stijgt: voor elke boom met 9 punten zijn er al minstens 1910 anti-magische labelingen. De

minimale kans om een juiste labeling te vinden, wordt echter snel kleiner (het aantal labelingen is $n!$, het aantal anti-magische labelingen neemt minder snel toe).

- * Een minder interessante constatering is het maximale aantal anti-magische labelingen bij een gegeven aantal punten. Dit blijkt namelijk altijd $n!$ te zijn, aangezien we een familie van bomen X_n kunnen maken met één punt x waarvoor $d(x) = n - 1$ en $n - 1$ punten y met $d(y) = 1$. Hieronder is een voorbeeld gegeven van deze boom met 7 punten (laten we deze de X_7 noemen):



- * Als we beter kijken naar de tabel met de kleinste aantallen, dan zien we het opvallende feit dat, voor elke van deze n 'en, de boom met de minste anti-magische labelingen een oude bekende is: het pad P_{n-1} (zie 2.1). Van deze familie van grafen weten we al dat ze anti-magisch zijn.

We zouden kunnen proberen een formule te bedenken voor het aantal goede labelingen voor de paden, om te kijken of we hier meer informatie uit kunnen halen. Daarom zien we in de volgende tabel de aantallen staan, met daarnaast hun priemfactorontbinding:

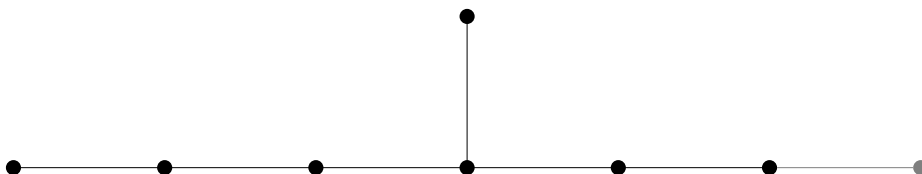
n	Aantal labelingen	Priemfactorontbinding
3	2	2
4	2	2
5	6	$2 \cdot 3$
6	16	2^4
7	92	$2^2 \cdot 23$
8	352	$2^5 \cdot 11$
9	1910	$2 \cdot 5 \cdot 191$
10	10892	$2 \cdot 7 \cdot 389$

Zoals we in de tabel kunnen zien, is er niet direct een verband te vinden tussen n en het aantal anti-magische labelingen. Hier heb ik dus verder geen conclusies uit kunnen trekken.

- * Als we alleen naar het grootste en het kleinste aantal kijken, zien we misschien minder van de relevante eigenschappen van deze grafen die ervoor zorgen dat er zo veel respectievelijk zo weinig goede labelingen zijn. Daarom ben ik ook gaan kijken naar de bomen die van alle overgebleven soorten de meeste en de minste anti-magische labelingen hebben (dus de op één na meeste en de op één na minste). Deze getallen zien we in de volgende tabel (vanwege het kleine aantal bomen met 3 en 4 punten zijn deze resultaten weinig inspirerend):

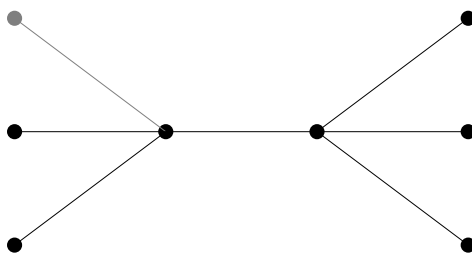
n	Weinig	Veel
5	12	12
6	30	96
7	142	648
8	596	4824
9	3226	38304

- * Zoals we eerder hebben gezien dat de kleinste aantallen behaald werden door de familie van de paden, zien we ook bij de kleinste aantallen van de overige bomen een duidelijk patroon: voor elke $n + 1$ vinden we hier een P_{n-1} met nog een extra punt aan een centroïde bevestigd. Hieronder zien we een afbeelding voor de boom met 8 punten. Wanneer het grijze gedeelte weggelaten wordt, zien we hoe de boom met 7 punten eruit ziet:



Het is duidelijk te zien dat deze bomen veel overeenkomsten vertonen met de paden. We zullen later proberen aan te geven welke eigenschappen beide bomen hebben die ervoor zorgen dat we weinig goede labelingen kunnen vinden. Eerst kijken we echter naar de familie van bomen met veel anti-magische labelingen.

- * Ook bij de bomen waar veel mogelijkheden zijn, is een duidelijk patroon te ontdekken. Ze zijn namelijk gelijk aan de eerder gedefinieerde bomen X_n , waarbij het middelste punt (met graad $n - 1$) is gesplitst in twee punten die aan elkaar verbonden zijn, terwijl alle andere punten gelijkmatig zijn verdeeld over beide punten (als n even; als n oneven heeft één van beide punten een buur extra). Hieronder zien we plaatje van deze bomen met 8 punten. Ook hier kunnen we het grijze gedeelte weglaten om de boom met 7 punten te verkrijgen:



- * Een mogelijke verklaring voor het verschil in aantal anti-magische labelingen tussen de bomen die we hier bekeken hebben, is het verschil in aantal symmetrieën van de bomen. Waar de bomen X_n een aantal van $(n - 1)!$ symmetrieën hebben, hebben de bomen P_n er slechts twee. Als de boom een bepaalde symmetrie heeft, dan kunnen we diezelfde symmetrie natuurlijk toepassen op een anti-magische labeling om zo een nieuwe, goede labeling te vinden.
- * Een andere verklaring voor het verschil in aantal ligt wellicht in de graden van de punten. Zoals we in paragraaf 2.5 gezien hebben, zijn grafen met een punt met hoge graad makkelijk anti-magisch te labelen (we konden voor veel

van de kanten een willekeurig label kiezen). Een punt van graad 1 zal ook niet snel problemen geven, aangezien alle labels verschillend zijn. Het vermoeden rijst dus dat grafen waarin het aantal punten met lage graad ongelijk aan 1 klein is, minder moeilijk zijn dan grafen waarbij dit aantal groot is.

- * Al met al kunnen we uit de resultaten van het experiment enkele aanknopingspunten vinden voor verder onderzoek, waarvoor ik in paragraaf 5.3 enkele suggesties geef.

5.2 Verbeterpunten

Wanneer je aan het eind van een experiment terugkijkt, is het bijna altijd zo dat er punten zijn die je liever anders (en hopelijk) beter had gedaan. In deze paragraaf zal ik hiervan enkele voorbeelden proberen te geven:

- * Ten eerste het programmeerwerk: in de beschreven algoritmes zit een aanzienlijk aantal stappen die voornamelijk bedoeld zijn om het overzicht te behouden. Een pakket als Magma zou in veel gevallen sneller het gewenste resultaat geven wanneer deze stappen anders geïmplementeerd waren. Daarnaast zijn er ongetwijfeld mogelijkheden om een sneller/beter algoritme te schrijven met functies/technieken die mij niet bekend zijn, of waaraan ik op dat moment niet gedacht heb.
- * Magma is een pakket dat bij uitstek geschikt is om wiskundige constructies in te maken. In deze algoritmes worden echter weinig specifieke wiskundige structuren gebruikt, waardoor de keuze voor een andere, snellere software wellicht beter was geweest.
- * Ook is het mogelijk dat er meer wiskundige eigenschappen van grafen, of labelingen te gebruiken zijn die een minder naïef idee dan alle labelingen nalopen mogelijk maakt.
- * Tijdens de zoektocht naar de anti-magische labelingen, moest het programma steeds alle gevonden labelingen bijhouden, terwijl we eigenlijk vooral geïnteresseerd waren in het aantal labelingen. Het ontbreken van deze geheugenlast had wellicht een stuk in de snelheid kunnen schelen.
- * De duur van het experiment (twee weken) is vrij willekeurig gekozen en heeft te maken met gunstige tijdplanningen. Wanneer het algoritme langer zou draaien (eventueel op een snellere computer), dan zouden er ongetwijfeld meer resultaten beschikbaar zijn, waaruit wellicht ook betere conclusies uit getrokken zouden kunnen worden.

5.3 Verdere zoekrichtingen

Naar aanleiding van dit onderzoek kunnen er verschillende vragen opgeworpen worden. Beantwoording van enkele hiervan zouden een stap zijn in de richting van een bewijs van het vermoeden. Enkele van zulke vragen zal ik hieronder proberen te formuleren:

- * In het vorige hoofdstuk hebben we gezien dat voor elke onderzochte n , het pad P_{n-1} de boom blijkt te zijn met de minste anti-magische labelingen. Wanneer bewezen zou kunnen worden dat dit voor elke n geldt, volgt hieruit automatisch dat elke boom anti-magisch te labelen is (aangezien we al gezien hebben

dat het pad altijd anti-magisch is). Een manier om dit te bewerkstelligen zou kunnen zijn een manier om een labeling van het pad over te voeren in een andere, willekeurige boom (dat ook altijd een pad bevat).

- * Wanneer gekeken wordt naar de automorfismegroep van een graaf, zijn er wellicht betere uitspraken te doen over het aantal anti-magische labelingen: misschien is er een schatting te bedenken voor de verhouding $\frac{\text{aantal anti-magische labelingen}}{\text{aantal automorfismen}}$. Een positieve onderschatting van deze verhouding zou weer automatisch een bewijs leveren voor het vermoeden.
- * Zoals in de vorige paragraaf even aangestipt is, zou het vinden van een formule voor het aantal labelingen voor het pad nuttig kunnen zijn. Wanneer deze eenmaal gevonden is, kan er wellicht ook een formule afgeleid worden voor andere klassen van grafen; wellicht zelfs de klasse van alle grafen.
- * Een andere mogelijkheid die onderzocht zou kunnen worden, is het samenvoegen van anti-magische labelingen, dat wil zeggen: een manier bij gegeven grafen (of bomen) G en H , met bijbehorende labelingen, een labeling te vinden voor een graaf waarin beide grafen verbonden zijn. Hier is verbonden een ruim begrip: hierbij kan gedacht worden aan een bipartite graaf waarbij in het ene stuk G te vinden is en in het andere H , maar ook zelfs aan de compositie van G en H (de graaf waar elk punt van G vervangen is door een exemplaar van H). Op deze manier zou een labeling vanuit kleinere grafen opgebouwd kunnen worden en zou een inductieargument het vermoeden kunnen bewijzen. Een kleinere, dus wellicht handelbare, versie van deze mogelijkheid is het toevoegen van één kant aan een anti-magische graaf en te bewijzen dat de graaf nog steeds anti-magisch is.
- * Een optie die directer aansluit op de uitgevoerde experimenten, is het bestuderen van de labelingen die daadwerkelijk anti-magisch blijken te zijn. Hebben deze nog bepaalde andere eigenschappen die op het eerste oog niet opvallen, maar waaruit conclusies getrokken kunnen worden die (een deel van) het vermoeden kunnen bevestigen?
- * Ten slotte is er natuurlijk ook nog de optie om een tegenvoorbeeld te zoeken: een graaf met een bewijs waarom die graaf niet anti-magisch is; danwel met een goed argument, danwel door alle labelingen na te gaan en te concluderen dat geen van allen voldoet.

6 Slot

In de afgelopen periode heb ik me bezig gehouden met een probleem dat in eerste instantie vrij eenvoudig leek (gezien de formulering), maar achteraf gezien toch moeilijk handelbaar is gebleken. Wanneer eenmaal een gevoel is ontstaan voor het maken van anti-magische labelingen, blijkt dat labelingen die werken bij bepaalde grafen, volledig misgaan bij vrijwel identieke grafen. Ook het feit dat het niet lukte om een labeling van een bepaalde graaf uit te breiden naar een labeling van die graaf met een extra kant, heeft ons doen besluiten om ons te beperken tot de (op het oog eenvoudigere) deelklasse van de bomen en op de experimentele toer te gaan. De database van bomen die ik daarmee gegenereerd heb, is een mooi resultaat, dat wellicht ook voor andere onderzoeken nuttig zou kunnen zijn. Ook de wetenschap dat deze uitgebreid zou kunnen worden in geval van beschikbaarheid van een snelle computer draagt bij aan de positieve kant van dit resultaat. Ook de database met alle gevonden labelingen zou een bron kunnen zijn die geheimen bezit voor eenieder die ze ziet.

Omdat dit onderzoek in het kader van een Bachelorscriptie voor drie studiepunten is gemaakt, is het vanzelfsprekend niet heel uitgebreid en zijn er weinig verheffende conclusies uit tevoorschijn gekomen. De mogelijkheden voor verder onderzoek zijn echter zeker aanwezig, waardoor dit stuk een basis kan vormen voor eenieder die een gooi wil doen naar het bewijzen van het vermoeden.

Tot besluit wil ik mijn begeleider, Dr. Wieb Bosma, bedanken voor de feedback en het meedenken in het proces dat tot deze scriptie heeft geleid.

Referenties

- [1] N. Hartsfield, G. Ringel, *Pearls in Graph Theory*, Dover Publications, Mineola, New York, 2003
- [2] D. Knuth, *The art of computer programming*, Volume 4 Fascicle 4, section 7.2.1.6 Generating All Trees, Addison-Wesley, 2006
- [3] N. Alon, G. Kaplan, A. Lev, Y. Roditty, R. Yuster, *Dense graphs are antimagic*, Journal of Graph Theory 47, 2004
- [4] W. Bosma, J. Cannon, C. Playoust, *The Magma Algebra System I: The User Language*, Journal of Symbolic Computation 24 (1997), 235–265
- [5] D. Hefetz, *Anti-magic graphs via the Combinatorial NullStellenSatz*, Journal of Graph Theory 50, 2005
- [6] Yongxi Cheng, *Lattice grids and prisms are antimagic*, Theoretical Computer Science Volume 374, 2007