



TECHNISCHE
UNIVERSITÄT
WIEN

DIPLOMARBEIT

Mustervermeidung in der Kombinatorik

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Masterstudium Technische Mathematik

eingereicht von

Gabriel Reikl B.Sc.

Matrikelnummer 01109467

ausgeführt am Institut für Diskrete Mathematik und Geometrie
der Fakultät für Mathematik und Geoinformation der Technischen Universität Wien

Betreuung

Betreuer: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Bernhard Gittenberger

Wien, 13.02.2019

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

Inhaltsverzeichnis

1	Mustervermeidung in Permutationen	6
2	Gitterpfade	8
2.1	Dyck-Pfade	8
2.1.1	Eine Verbindung zu den 123 vermeidenden Permutationen	9
2.2	Peaks	10
2.3	Peaks eingeschränkt auf eine bestimmte Menge	13
2.3.1	Peaks auf geraden oder ungeraden Höhen	16
2.4	Täler	17
2.5	Klassische Mustervermeidung	19
2.6	Mustervermeidung in Gitterpfaden	22
3	Bäume	29
3.1	Heaps	29
3.1.1	Binäre Strauchwälder	30
3.1.2	Binäre Heaps	32
3.1.3	k-Heaps	36
3.2	Markierte Bäume	37
3.2.1	Die Cluster Inversions Formel	39
3.2.2	Asymptotisches Verhalten	41
3.3	Binärbäume	42
3.4	k - Bäume	50
3.5	Unzusammenhängende Teilbäume	53
3.6	Unbalancierte Teilbäume	58
4	Catalan-Strukturen	62
4.1	Catalan-Folgen	62
4.1.1	Die konstante Folge der Länge k	63
4.2	Modulare Catalan-Zahlen	65
4.2.1	k-Assoziativität	65
4.2.2	Binärbäume	66
4.2.3	Ebene Bäume	67
4.2.4	Hauptsatz über die modularen Catalan-Zahlen	70
4.2.5	Die verallgemeinerten Motzkin-Zahlen	72
4.2.6	Explizite Berechnung	74
5	Verschiedenes	75
5.1	Strings	75

Inhaltsverzeichnis

5.2	Verallgemeinerung der Mustervermeidung	76
5.2.1	Variable Muster	76
5.2.2	Mustervermeidung auf Graphen	77
5.3	Mengenpartitionen	77
6	Anwendungen	80
6.1	Informatik	80
6.2	Biologie	80
6.3	Praktische Anwendung	80

Einleitung

Das Ziel dieser Arbeit ist es, Mustervermeidung in verschiedensten kombinatorischen Strukturen zu betreiben. Wir beginnen bei den Gitterpfaden, danach sind verschiedenste Arten von Bäumen an der Reihe, aber auch Strings, Graphen und Mengenpartitionen werden wir behandeln. Das Thema Mustervermeidung stammt ursprünglich aus dem Gebiet der Permutationen, welche in der Literatur den größten Anteil der Mustervermeidung ausmacht. Wir werden dieses Thema im ersten Kapitel streifen um einige wichtige Erkenntnisse festzuhalten. Der Fokus dieser Arbeit soll jedoch auf den Kapiteln danach liegen, wo wir versuchen werden die Mustervermeidung auf andere Strukturen zu übertragen.

Die Mustervermeidung setzt sich aus einigen verschiedenen Punkten zusammen. Der erste und wesentlichste Punkt ist die Frage nach der Definition des Musters. Es ist zum Beispiel im vorhinein nicht immer klar, was ein Muster für eine beliebige kombinatorische Struktur ist und wann es enthalten wird. Für Permutationen sind es kleinere Permutationen. Jedoch, wie wir im ersten Kapitel feststellen werden, muss man im vorhinein festlegen ob es sich um ein zusammenhängendes Muster handelt oder nicht. Ein ähnliches Problem tritt auch im zweiten Kapitel über Gitterpfade auf. Hier betrachten wir einerseits zusammenhängende Muster, aber auch unzusammenhängende. Im dritten Kapitel bearbeiten wir verschiedenste Arten von Bäumen. Hier wird der Fall auftreten, dass unter anderem Permutationen als Muster auftreten, also nicht einmal die gleiche kombinatorische Struktur. Im vierten Kapitel beschäftigen wir uns mit den verschiedenen Verbindungen zwischen den einzelnen Strukturen und im fünften Kapitel werden wir verschiedene Themen behandeln um einen Ausblick zu geben, was es noch an offenen Möglichkeiten gibt. Schlussendlich werden wir uns mit den Anwendungen der Mustervermeidung beschäftigen.

Hat man die Definition eines Musters und wann es enthalten ist, so kann man mit der eigentlichen Arbeit beginnen. Der einfachste Ansatz, ist das pattern matching. Hier geht es darum für ein festes Objekt, Vorkommen eines Musters zu finden. Hierfür gibt es vor allem in der Informatik zahlreiche Anwendungen. Wir werden es in dieser Arbeit jedoch nicht weiter behandeln. Für uns wird das erste Ziel sein simpelste Muster zu vermeiden, das sind zum Beispiel Permutationen der Länge zwei und drei. Ist dies geklärt, so werden wir versuchen die erzeugende Funktion für Strukturen, welche ein beliebiges Muster vermeiden, zu finden. Der nächste logische Schritt ist es, mehrere Muster zu vermeiden. Sind diese Punkte erledigt, so werden wir versuchen das Muster nicht mehr zu vermeiden, sondern fix vorgeben, wie oft es enthalten sein soll. Hier ist das Ziel eine bivariate

Inhaltsverzeichnis

erzeugende Funktion zu finden, wobei die eine Variable die Zahl der Strukturen zählt und die andere die Zahl der Vorkommnisse. Sind all diese Punkte erledigt, so werden wir versuchen Aussagen über das asymptotische Verhalten zu treffen.

Im Verlauf dieser Arbeit werden wir einige verschiedene Muster antreffen, welche im Großen und Ganzen alle nach diesem Schema abgearbeitet werden. Ein gutes Beispiel hierfür sind die Peaks aus dem Kapitel Gitterpfade. Dieses Muster wird relativ gut nach dem obigen Schema behandelt. Jedoch werden wir auch auf Muster treffen, wo dies aus verschiedensten Gründen nicht möglich ist, beispielsweise bei den unbalancierten Binärbäumen.

1 Mustervermeidung in Permutationen

Historisch gesehen wurde die Mustervermeidung zuerst im Gebiet der Permutationen betrieben. Dies ist ein Grund dafür, dass wir in diesem Bereich schon einiges an Theorie und Werkzeugen besitzen, jedoch gibt es auch noch einige offene Fragen. Thema dieser Arbeit ist die Mustervermeidung auf anderen kombinatorischen Strukturen, wir benötigen aber immer wieder Hilfsmittel, welche für die Permutationen entwickelt wurden, welche wir hier anführen wollen.

Definition 1.1. Die Menge der Permutationen der Länge n wird mit \mathcal{S}_n bezeichnet und es gilt

$$|\mathcal{S}_n| = n!.$$

Bemerkung 1.2. Sei $\pi = \pi_1\pi_2\dots\pi_n$ eine Permutation, so ist auch $\pi^r = \pi_n\pi_{n-1}\dots\pi_2\pi_1$ eine Permutation, genannt die reverse Permutation. Ebenso ist auch das Komplement $\pi^c = (n+1-\pi_1)(n+1-\pi_2)\dots(n+1-\pi_n)$ wieder eine Permutation der Länge n .

Nun kommen wir zur Mustervermeidung von Permutationen. Hier gilt, dass Muster wieder Permutationen kürzerer Länge sind.

Definition 1.3. Sei $\pi = \pi_1\pi_2\dots\pi_n$ eine Permutation und $p = p_1p_2\dots p_k$ ein Muster der Länge $k \leq n$. Die Permutation π enthält genau dann das Muster p , falls es Indizes $1 \leq i_1 \leq \dots \leq i_k \leq n$ gibt, sodass $\pi_{i_a} < \pi_{i_b}$ gilt, genau dann wenn $p_a < p_b$ gilt. Ansonsten sprechen wir davon, dass π das Muster p vermeidet.

Bemerkung 1.4. Eine Permutation π enthält genau dann ein Muster p , falls es eine unzusammenhängende Teilfolge von π gibt, welche ordnungsisomorph zu p ist.

Wir bezeichnen die Menge aller Permutationen der Länge n , welche das Muster p vermeiden mit $\mathcal{S}_n(p)$.

Beispiel 1.5. Die Mächtigkeit von $\mathcal{S}_n(213)$ entspricht genau den Catalan-Zahlen C_n . Desweiteren gilt für jede Permutation p der Länge 3, dass

$$|\mathcal{S}_n(p)| = C_n$$

gilt.

Desweiteren werden Permutationen der Länge n , welche eine Menge von Mustern $P = \{p_1, p_2, \dots\}$ vermeiden, mit $\mathcal{S}_n(P)$ oder mit $\mathcal{S}_n(p_1, p_2, \dots)$ bezeichnet.

1 Mustervermeidung in Permutationen

Beispiel 1.6. Beispiele dafür sind die Permutationen welche die Muster 213 und 231 vermeiden, beziehungsweise 213 und 312. Denn es gilt

$$\mathcal{S}_n(213, 231) = \mathcal{S}_n(213, 312) = 2^{n-1}$$

Schlussendlich kommen wir zu einer zweiten Definition, welche seltener verwendet wird. Hier sprechen wir von einem zusammenhängenden Muster.

Definition 1.7. Sei $\pi = \pi_1\pi_2 \dots \pi_n$ eine Permutation und $p = p_1 - p_2 - \dots - p_k$ ein zusammenhängendes Muster. (An den Bindestrichen zwischen den Eintägen erkennbar.) So enthält π das Muster p , falls es einen Index i gibt, sodass die Einträge $\pi_i\pi_{i+1} \dots \pi_{i+k-1}$ ordnungsisomorph zu p sind.

Beispiel 1.8. So bezeichnet $\mathcal{S}_n(1 - 2 - \dots - k)$ die Menge aller Permutationen der Länge n , welche keine zusammenhängende aufsteigende Teilfolge der Länge k besitzen.

Das nächste Konzept, welches wir öfter benötigen werden ist die Wilf-Äquivalenz.

Definition 1.9. Zwei Permutationen p, q der Länge k , werden Wilf-äquivalent genannt, falls für alle n

$$|\mathcal{S}_n(p)| = |\mathcal{S}_n(q)|$$

gilt.

Bemerkung 1.10. Offensichtlich wird durch die obige Definition eine Äquivalenzrelation auf der Menge der Permutationen der Länge n definiert.

Ein weiterer wichtiger Punkt, welcher mit der Mustervermeidung zusammenhängt, ist das Zählen, wie oft ein Muster vorkommt. So werden zum Beispiel für eine Permutation p die Vorkommnisse des Musters 21 gezählt, also die Zahl der Fehlstände.

Definition 1.11. Zwei Permutationen p, q werden stark Wilf-äquivalent genannt, falls für jedes k die Zahl der Permutation, welche das Muster p genau k mal enthalten, gleich der Zahl der Permutationen, welche das Muster q genau k mal enthalten, ist.

2 Gitterpfade

Unser erstes eigentliches Kapitel trägt den Titel Gitterpfade, da wir uns, beginnend mit den Dyck-Pfaden, mit den verschiedensten Arten von Gitterpfaden auseinandersetzen werden. Zusätzlich dazu werden wir auch mit vielen verschiedenen Arten von Mustern in Berührung kommen. Unser erstes Muster werden die Peaks eines Dyck-Pfads sein, welche wir wie in der Einleitung beschrieben vollständig bearbeiten werden. Dies soll auch repräsentativ dafür sein, wie das Ergebnis aussehen sollte, wenn man sich eingehend mit einem Thema befasst.

2.1 Dyck-Pfade

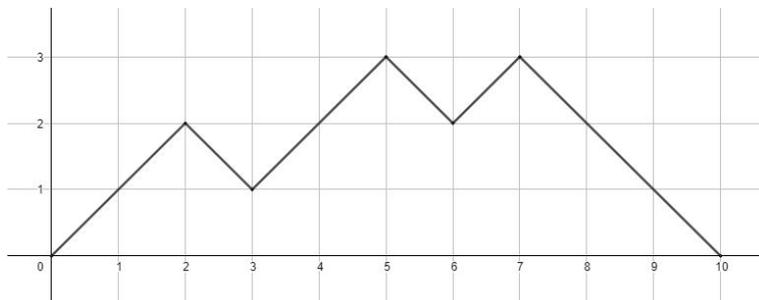
Unsere Arbeit beginnt mit einer vertrauten kombinatorischen Struktur, den Dyck-Pfaden. Wir wollen zuerst einige wesentliche Eigenschaften wiederholen, bevor wir uns in die Mustervermeidung stürzen, vgl. [2].

Definition 2.1. Ein Dyck-Pfad ist ein Gitterpfad mit Startpunkt $(0, 0)$ und Endpunkt $(2n, 0)$, welcher aus zwei verschiedenen Arten von Schritten besteht: $U = (1, 1)$, auch Up-Step genannt und $D = (1, -1)$ oder auch Down-Step. Dabei darf der Pfad niemals die x-Achse unterschreiten. Die Länge des Pfades ist definiert als die Summe der Up und Down-Steps. Ein Pfad der Länge $2n$ wird auch n-Dyck-Pfad genannt. Die Menge aller n-Dyck-Pfade wird mit \mathcal{D}_n bezeichnet und \mathcal{D} beschreibt die Menge aller Dyck-Pfade.

Bemerkung 2.2. Ein Dyck-Pfad kann auch durch sein zugehöriges Dyck-Wort eindeutig beschrieben werden. Ein Dyck-Wort ist eine endliche Abfolge aus dem Alphabet $\{U, D\}$, wobei folgende zwei Bedingungen erfüllt werden müssen: Erstens müssen gleich viele Us wie Ds im Dyck-Wort vorkommen. Zweitens muss vor dem i-tem D bereits i Us stehen.

Bemerkung 2.3. Jeder n-Dyck-Pfad kann auch als ein Klammersausdruck für $n + 1$ Variable interpretiert werden.

Beispiel 2.4. Wir betrachten einen Dyck-Pfad der Länge 10, mit dem dazugehörigen Dyck-Wort $UUDUUDUDDD$.



2 Gitterpfade

Abbildung 1

Desweiteren werden die n-Dyck-Pfade durch die gut bekannten Catalan-Zahlen C_n gezählt, welche durch

$$C_n := \frac{1}{n+1} \binom{2n}{n}$$

definiert sind. Die erzeugende Funktion der Dyck-Pfade wird mit

$$C(x) = \frac{1 - \sqrt{1 - 4x}}{2x}$$

bezeichnet und es gilt die Gleichung

$$C(x) = 1 + xC(x)^2. \quad (1)$$

2.1.1 Eine Verbindung zu den 123 vermeidenden Permutationen

Da die Menge der n-Dyck-Pfade und die Permutationen welche das Muster 123 vermeiden gleichmächtig sind, gibt es klarerweise Bijektionen zwischen diesen beiden Strukturen. Wir werden in diesem Abschnitt die natürliche Abbildung Ψ kennenlernen, welche keine willkürliche Bijektion zwischen beiden Strukturen ist, sondern gewisse Eigenschaften erhält, welche später noch von Nutzen sind, siehe [1].

Definition 2.5. Die Abbildung $\Psi : \mathcal{S}_n(123) \rightarrow \mathcal{D}_n$ ist definiert als jene Abbildung, welche der Permutation $\pi = \pi_1 \dots \pi_n$ folgenden Dyck-Pfad d der Länge n zuordnet. Seien m_1, \dots, m_s die Rechts-Links-Maxima der Permutation π , sodass wir sie in der Form

$$\pi = w_s m_s w_{s-1} m_{s-1} \dots w_1 m_1$$

darstellen können. Die Teilfolgen zwischen den Maxima werden mit w_i bezeichnet. Nun entspricht das Bild von Ψ folgendem Dyck-Pfad. Wir lesen die obige Darstellung von links nach rechts und jedes w_i entspricht $(|w_i| + 1)$ Up-Steps und jedes Maximum entspricht $(m_i - m_{i-1})$ Down-Steps (wobei $m_0 = 0$ gilt).

Bemerkung 2.6. Die Teilfolgen w_i müssen monoton fallend sein, da jeder Fehlstand der Permutation genau einem m_i entsprechen muss. Desweiteren müssen alle Elemente von w_i größer als das Maximum von w_{i+1} sein.

Lemma 2.7. Die oben definierte Abbildung Ψ ist wohldefiniert und eine Bijektion.

2 Gitterpfade

Beweis. Für die Wohldefiniertheit müssen wir nach Bemerkung 2.2 zeigen, dass es genau n Up / Down-Steps gibt und vor dem i -ten Down-Step bereits i Up-Steps gemacht wurden. Durch die Definition von Ψ gibt es $\sum_i m_i - m_{i-1} = m_s = n$ Down-Steps. Ebenso gibt es für jede Zahl der Permutation einen Up-Step, denn jeder der Blöcke $w_i m_i$ besteht aus $(|w_i| + 1)$ Zahlen. Nach der gleichen Idee kann man sich überlegen, dass der Term $m_i - m_{i-1}$ den $(n - m_{i-1})$ -ten Down-Step erzeugt. Jedoch muss davor für jede der Zahlen größer m_{i-1} ein Up-Step gemacht worden sein, also mehr als $n - m_{i-1}$.

Die Bijektivität zeigen wir indem wir uns überlegen, dass die Abbildung injektiv sein muss. Da zusätzlich beide Mengen gleich groß sind, muss es sich um eine Bijektion handeln. \square

2.2 Peaks

Als Grundlage für diesen Abschnitt dient [2].

Nun da wir uns mit der ersten kombinatorischen Struktur auseinandergesetzt haben, kommen wir zu unserem ersten Muster.

Definition 2.8. Ein Peak ist ein Punkt auf einem Dyck-Pfad, welcher direkt nach einem Up-Step und direkt vor einem Down-Step liegt. Die Höhe eines Peaks bezeichnet seine y-Koordinate.

Bemerkung 2.9. Ein Peak nach dieser Definition entspricht einem zusammenhängenden Muster bestehend aus einem Up-Step und einem Down-Step.

Klarerweise muss jeder nicht triviale Dyck-Pfad mindestens einen Peak enthalten.

Bemerkung 2.10. Es ist bekannt, dass die Zahl der n -Dyck-Pfade, mit genau k Peaks, mit den Narayana Zahlen $N(n, k)$ übereinstimmt, welche durch

$$N(n, k) := \frac{1}{n} \binom{n}{k} \binom{n}{k-1} \quad (2)$$

definiert sind. Nach dieser Interpretation muss klarerweise

$$\sum_{k=1}^n N(n, k) = C_n$$

gelten.

Um unsere interesse an den Peaks noch etwas zu motivieren, bemühen wir die in Definition 2.5 beschriebene Abbildung Ψ . Diese hat die Eigenschaft Peaks auf bestimmte Muster abzubilden, was folgendes Lemma zeigt.

2 Gitterpfade

Lemma 2.11. *Sei $\pi = \pi_1 \dots \pi_n$ eine Permutation welche das Muster 123 vermeidet und sei $D = \Psi(\pi)$ der korrespondierende Dyck-Pfad. Ein Peak von D auf Höhe i entspricht genau einem Element π_j , welches das letzte Element eines Vorkommnis des Musters $(i-1)(i-2) \dots 21i$ ist.*

Beweis. Ein Peak im Dyck-Pfad entspricht nach der Definition einem Rechts-Links-Maximum m , wobei die Höhe von m der Zahl Elemente kleiner m entspricht. Diese Elemente können kein Rechts-Links-Maximum bilden und müssen daher innerhalb der w_i liegen. Nach Bemerkung 2.6 müssen diese Elemente in absteigender Reihenfolge angeordnet sein. Zusammen mit dem Element m bilden sie das gewünschte Muster. □

Wir werden im weiteren Verlauf Dyck-Pfade mit Peaks auf einer bestimmten Höhe behandeln. Dabei handelt es sich nicht um Mustervermeidung im gewöhnlichen Sinn, sondern wir betrachten eine zusätzliche Struktur. Es stellt sich die Frage: Wie viele Dyck-Pfade ohne Peaks auf Höhe k gibt es?

Dafür benötigen wir jedoch noch etwas an Notation. Sei $p(m, n)$ die Zahl der Dyck-Pfade der Länge $2n$ welche keine Peaks auf Höhe m haben. Die dazugehörige erzeugende Funktion $\text{Peak}_m(x) = \sum_{n \geq 0} p(m, n)x^n$ beschreibt die Menge aller Dyck-Pfade ohne Peaks auf Höhe m . Nun können wir unseren ersten Satz dieses Kapitels formulieren, welcher die Dyck-Pfade ohne Peak auf Höhe 1 beschreibt.

Satz 2.12. *Es gilt*

$$\text{Peak}_1(x) = \sum_{n=0}^{\infty} p(1, n)x^n = \frac{1}{1 - x^2 C(x)^2}. \quad (3)$$

Beweis. Wir zeigen die Gleichung, indem wir einen beliebigen Dyck-Pfad der Länge $2n$ ohne Peak auf Höhe 1 konstruieren. Unser Pfad muss mit einem Up-Step beginnen, gefolgt von einem beliebigen Dyck-Pfad der Länge $2k$. Dadurch kann es zu keinem Peak auf Höhe 1 kommen. Darauf folgt ein Down-Step und ein beliebiger Dyck-Pfad ohne Peak auf Höhe 1. Diese Konstruktion ist ausreichend, um eine Gleichung aufzustellen. Für $n \geq 2$ können wir über alle k zwischen 1 und $n-1$ aufsummieren und wir erhalten

$$p(1, n) = \sum_{k=1}^{n-1} p(1, n-k-1)C_k.$$

Aus dem Cauchy Produkt $\text{Peak}_1(x)C(x) = \sum_{k=0}^n C_k p(1, n-k)$ können wir fol-

2 Gitterpfade

gern, dass

$$\begin{aligned} \sum_{k=1}^{n-1} p(1, n-k-1)C_k &= [x^{n-1}](C(x)\text{Peak}_1(x)) - p(1, n-1) \\ &= [x^n](xC(x)\text{Peak}_1(x)) - p(1, n-1) \end{aligned}$$

gilt. Daraus und durch die Tatsache, dass $p(1, 0) = 1$ und $p(1, 1) = 0$ gelten, folgt

$$\begin{aligned} \text{Peak}_1(x) &= 1 + \sum_{n \geq 2} p(1, n)x^n = 1 + xC(x)\text{Peak}_1(x) - x - x\text{Peak}_1(x) + x \\ &= 1 + x\text{Peak}_1(x)(C(x) - 1) = 1 + x\text{Peak}_1(x)xC(x)^2 \end{aligned}$$

Durch einfaches Umformen ergibt sich dann

$$\text{Peak}_1(x) = \frac{1}{1 - x^2C(x)^2}.$$

□

Bemerkung 2.13. Die erzeugende Funktion $\text{Peak}_1(x)$ entspricht der erzeugenden Funktion der Fine-Zahlen $F(x) = \sum_{n \geq 0} f_n x^n$ (Siehe Appendix 6.3).

Nun da wir den ersten Schritt erledigt haben, können wir rekursiv die erzeugende Funktion $\text{Peak}_m(x)$ für alle m herleiten.

Satz 2.14. Für $m \geq 2$ gilt

$$\text{Peak}_m(x) = \frac{1}{1 - x\text{Peak}_{m-1}(x)}. \quad (4)$$

Beweis. Der Beweis dieses Satzes erfolgt analog zu dem von Satz 2.12. Ein beliebiger Dyck-Pfad ohne Peak auf Höhe k beginnt mit einem Up-Step, gefolgt von einem Pfad der Länge $2j$ ohne Peak auf Höhe $k-1$. Darauf folgen ein Down-Step und ein Dyck-Pfad der Länge $2n - 2j - 2$ ohne Peak auf Höhe k . Nun können wir über alle j von 0 bis $n-1$ aufsummieren und erhalten

$$p(m, n) = \sum_{j=0}^{n-1} p(m-1, j)p(m, n-1-j) = [x^{n-1}](\text{Peak}_{m-1}(x)\text{Peak}_m(x)).$$

Klarerweise gilt $p(m, 0) = 1$ und für $n \geq 1$ folgt

$$\text{Peak}_m(x) = 1 + x\text{Peak}_{m-1}(x)\text{Peak}_m(x),$$

und nach kurzem Umformen erhält man

$$\text{Peak}_m(x) = \frac{1}{1 - x\text{Peak}_{m-1}(x)}.$$

□

2 Gitterpfade

Korollar 2.15. *Die Zahl der n -Dyck-Pfade ohne Peak auf Höhe 2 ist gleich der Catalan-Zahlen C_{n-1} und es gilt*

$$\text{Peak}_2(x) = 1 + xC(x).$$

Beweis. Wir benötigen Satz 2.14 und Gleichung (1) für folgende Umformungen:

$$\begin{aligned} \text{Peak}_2(x) &= \frac{1}{1 - x\text{Peak}_1(x)} = \frac{1}{1 - \frac{x}{1-x^2C(x)^2}} \\ &= \frac{1}{1 - \frac{x}{1+x-xC(x)}} = \frac{1+x-xC(x)}{1-xC(x)} \\ &= 1 + xC(x) \end{aligned}$$

□

Wir beenden dieses Unterkapitel mit einer Aussage über das asymptotische Verhalten von Dyck-Pfaden ohne Peaks auf Höhe 1 beziehungsweise 2.

Proposition 2.16. *Es gelten folgende beiden Aussagen*

i)

$$\lim_n \frac{f_n}{C_n} = 4/9,$$

ii)

$$\lim_n \frac{C_{n-1}}{C_n} = 1/4.$$

Das bedeutet, dass für n ausreichend groß, ca. $4/9$ der Dyck-Pfade keine Peaks auf Höhe 1 haben und ca. $1/4$ der Dyck-Pfade haben keine Peaks auf Höhe 2.

2.3 Peaks eingeschränkt auf eine bestimmte Menge

Als nächstes verallgemeinern wir das bisher Erarbeitete und werden Dyck-Pfade betrachten, welche nur Peaks innerhalb einer bestimmten Menge haben, beziehungsweise deren Peaks eine bestimmte Menge an Höhen vermeiden. Aussagen dieses Unterkapitels stammen aus [4]. Zunächst wollen wir Satz 2.14 verallgemeinern, benötigen hierfür jedoch noch etwas Notation. Sei d ein Dyck-Pfad, dann bezeichnet P_d die Menge aller Peaks des Pfades d . Für $A \subseteq \mathbb{N}$, bezeichnet $\text{Peak}_A(x)$ die erzeugende Funktion der Dyck-Pfade d , für die $A \cap P_d = \emptyset$ gilt. Die Menge $A-1$ ist durch $\{n-1 : n \in A\} \cap \mathbb{N}$ definiert und β bezeichnet die Boolesche Funktion $\beta(\text{wahr}) = 1$ und $\beta(\text{falsch}) = 0$.

2 Gitterpfade

Satz 2.17. *Es gilt*

$$Peak_A(x) = \frac{1}{1 + x\beta(1 \in A) - xPeak_{A-1}(x)}. \quad (5)$$

Beweis. Analog zum Beweis von Satz 2.12 unterteilen wir auch hier den Dyck-Pfad. Es beginnt ein Up-Step, darauf folgt ein beliebiger k -Dyck-Pfad, dessen Peaks nicht in der Menge $A - 1$ liegen dürfen. Danach kommt ein Down-Step gefolgt von einem $(n - k - 1)$ -Dyck-Pfad, dessen Peaks nicht in der Menge A liegen dürfen. Es bezeichnet $p_A(n)$ die Zahl der n -Dyck-Pfade, deren Peaks nicht in der Menge A liegen. Es gilt für $n \geq 1$

$$p_A(n) = \sum_{k=0}^{n-1} p_{A-1}(k)p_A(n - k - 1) - \beta(1 \in A)p_A(n - 1).$$

Wir müssen den β Term hinzufügen, da durch den leeren Pfad ein Peak auf Höhe 1 entstehen könnte, welcher möglicherweise nicht erlaubt ist. Damit erhalten wir Folgendes:

$$\begin{aligned} p_A(n) &= [x^{n-1}](Peak_{A-1}(x)Peak_A(x) - \beta(1 \in A)Peak_A(x)) \\ &= [x^n]x(Peak_{A-1}(x) - \beta(1 \in A))Peak_A(x). \end{aligned}$$

Es gilt $p_A(0) = 1$ und durch Aufschreiben als formale Potenzreihe

$$Peak_A(x) = \sum_{n \geq 0} p_A(n)x^n = 1 + x(Peak_{A-1}(x) - \beta(1 \in A))Peak_A(x)$$

erhalten wir eine Gleichung, aus der durch kleine Umformungen das Gewünschte folgt.

□

Es gilt $Peak_{\emptyset}(x) = C(x)$. Für eine endliche Menge A und mithilfe des obigen Satzes lässt sich für die Funktion $Peak_A(x)$ immer explizit darstellen.

Beispiel 2.18. Sei $A = \{2,3\}$, dann erhalten wir durch Satz 5 die Gleichung

$$Peak_A(x) = \frac{1}{1 - \frac{x}{1+x - \frac{x}{1+x-xC(x)}}}.$$

Durch Umformungen mit Gleichung (1) erhalten wir schlussendlich $Peak_A(x) = 1 + x + x^2C(x)$. Das bedeutet, dass die Dyck-Pfade ohne Peak auf Höhe 2 beziehungsweise 3 durch die um 2 versetzten Catalan-Zahlen C_{n-2} gezählt werden.

2 Gitterpfade

Bemerkung 2.19. Das letzte Beispiel und die bereits in Korollar 2.15 bewiesene Aussage bewegten die Autoren von [4] folgenden Satz zu formulieren.

Satz 2.20. *Seien $m, n \in \mathbb{N}$ für die gilt $n \geq m \geq 2$, dann ist die Zahl der n -Dyck-Pfade, welche keine Peaks innerhalb der Menge $\{2, 3, \dots, m\}$ haben, gleich den versetzten Catalan-Zahlen C_{n-m+1} .*

Der Beweis dazu wird im nächsten Unterkapitel nachgebracht, da er dort direkt aus einem Satz folgt.

Als nächstes kommen wir zu den Dyck-Pfaden, welche nur Peaks innerhalb einer bestimmten Menge haben. Die entsprechende erzeugende Funktion wird mit $\text{Peak}_{\bar{R}}(x)$ bezeichnet.

Satz 2.21. *Für $R \subseteq \mathbb{N}$ gilt*

$$\text{Peak}_{\bar{R}}(x) = \frac{1}{1 + x\beta(1 \notin R) - x\text{Peak}_{\bar{R-1}}(x)}. \quad (6)$$

Für eine endliche Menge kann diese Funktion natürlich wieder explizit dargestellt werden, wenn man bedenkt, dass $\text{Peak}_{\bar{\emptyset}}(x) = 1$ gilt.

Beweis. Dieser Beweis folgt dem üblichen Schema. Es bezeichnet $d_R(n)$ die Menge aller Dyck-Pfade der Länge $2n$, welche Peaks innerhalb der Menge R haben. Ein beliebiger Pfad beginnt mit einem Up-Step, darauf folgt ein k -Dyck-Pfad, welcher nur Peaks in der Menge $R - 1$ hat. Darauf folgt ein Down-Step und schlussendlich ein $(n - k - 1)$ -Dyck-Pfad, welcher nur Peaks in der Menge R hat. Also gilt für $n \geq 1$

$$d_R(n) = \sum_{k=0}^{n-1} d_{R-1}(k)d_R(n - k - 1) - \beta(1 \notin R)d_R(n - 1).$$

Der letzte Summand muss wieder hinzugefügt werden, da für den Fall $k = 0$ ein Problem auftritt, falls die Zahl 1 nicht in R enthalten ist. Denn klarerweise ist der triviale Pfad in $d_R(0)$ enthalten. Nach der Konstruktion könnte dies aber einen Peak auf Höhe 1 erzeugen. Nun können wir obigen Ausdruck wieder als Koeffizienten einer anderen Potenzreihe anschreiben und erhalten

$$\begin{aligned} d_R(n) &= [x^{n-1}](\text{Peak}_{\bar{R-1}}(x)\text{Peak}_{\bar{R}}(x) - \beta(1 \notin R)\text{Peak}_{\bar{R}}(x)) \\ &= [x^n]x(\text{Peak}_{\bar{R-1}}(x) - \beta(1 \notin R))\text{Peak}_{\bar{R}}(x). \end{aligned}$$

Als Potenzreihe aufgeschrieben und mit der Tatsache, dass $d_R(0) = 1$ gilt

$$\text{Peak}_{\bar{R}}(x) = \sum_{n \geq 0} d_R(n) = 1 + x(\text{Peak}_{\bar{R-1}}(x) - \beta(1 \notin R))\text{Peak}_{\bar{R}}(x).$$

2 Gitterpfade

Der Rest folgt aus simplen Umformungen.

□

2.3.1 Peaks auf geraden oder ungeraden Höhen

Ziel dieses Abschnitts ist, Dyck-Pfade zu untersuchen, welche Peaks auf geraden, beziehungsweise ungeraden Höhen vermeiden. Hierfür benötigen wir jedoch etwas Wissen über Motzkin-Pfade.

Definition 2.22. Ein *Motzkin-Pfad* ist ein Gitterpfad mit Startpunkt $(0, 0)$ und Endpunkt $(n, 0)$, wobei drei verschiedene Arten von Schritten erlaubt sind: $U = (1, 1)$, $D = (1, -1)$ und ein so genannter Level-Step $L = (1, 0)$. Die x-Achse darf hierbei jedoch nicht unterschritten werden. Ein solcher Pfad wird auch n-Motzkin-Pfad genannt und die Menge aller n-Motzkin-Pfade wird mit \mathcal{M}_n bezeichnet.

Bemerkung 2.23. Es ist wichtig zu bemerken, dass es sich bei Motzkin-Pfaden nicht um Gitterpfade im klassischen Sinn handelt.

Die Zahlen $M_n = |\mathcal{M}_n|$ sind als Motzkin-Zahlen bekannt. Die Funktion $M(x)$ bezeichnet die entsprechende erzeugende Funktion. Es gilt

$$M(x) = 1 + xM(x) + x^2M(x)^2 = \sum_{n \geq 0} M_n x^n. \quad (7)$$

Noch eine weitere Familie an Zahlen hat in diesem Kontext Bedeutung. Die Riordan Zahlen R_n entsprechen der Mächtigkeit der Menge der Motzkin-Pfade, welche keinen Level-Step auf der x-Achse machen.

Satz 2.24. Für $n \geq 1$ ist die Zahl der n-Dyck-Pfade, welche keine Peaks innerhalb der geraden Zahlen haben, gleich der versetzten Motzkin-Zahlen M_{n-1} .

Beweis. Wir zeigen den Satz, indem wir eine Bijektion Φ zwischen der Menge der (n-1)-Motzkin-Pfade und den n-Dyck-Pfaden ohne Peak auf gerader Höhe, aufstellen. Sei $\phi : \{U, L, D\} \rightarrow \{UU, DU, DD\}$ die Abbildung, für die $\phi(U) = UU$, $\phi(L) = DU$ und $\phi(D) = DD$ gilt. Sei $M = M_1 M_2 \dots M_{n-1}$ ein Motzkin-Pfad, dann ist $\phi(M)$ definiert als $\phi(M_1)\phi(M_2)\dots\phi(M_{n-1})$. So definieren wir $\Phi(M) := U\phi(M)D$. Das Bild von Φ kann klarerweise nur Peaks auf ungeraden Höhen haben. Damit ist Φ offensichtlich eine Bijektion.

□

2 Gitterpfade

Satz 2.25. *Sei $n \geq 1$, dann ist die Zahl der n -Dyck-Pfade, welche keinen Peak auf ungeraden Höhen haben, gleich den Riordan Zahlen R_n .*

Beweis. Da die Abbildung ϕ aus dem letzten Beweis eine Bijektion ist, können wir die Abbildung ϕ^{-1} betrachten. Diese bildet n -Dyck-Pfade mit Peaks auf geraden Höhen auf n -Motzkin-Pfade ab. Da es aber keinen Peak auf Höhe 0 geben kann, gibt es im Bild von ϕ keinen Pfad mit Level-Step auf Höhe 0.

□

2.4 Täler

Nun, da wir Dyck-Pfade und Peaks ausreichend beschrieben haben, kommen wir zum nächsten Thema. Die nächste natürlich auftretende Fragestellung betrifft die Zahl der Dyck-Pfade mit einer fest vorgegebenen Anzahl an Peaks auf einer bestimmten Höhe. Dafür benötigen wir jedoch Theorie, aus dem Paper [3] von Toufik Mansour, über ein zweites sehr ähnliches Muster, welches noch enger mit den Peaks verbunden ist, als auf den ersten Blick sichtbar.

Definition 2.26. Ein Tal ist ein Punkt auf einem Dyck-Pfad, welcher direkt nach einem Down-Step und direkt vor einem Up-Step liegt. Die Höhe des Tals wird analog zu den Peaks durch seine y -Koordinate beschrieben.

Die erzeugende Funktion der Dyck-Pfade, welche genau r Täler auf Höhe k haben, bezeichnen wir mit $\text{Valley}_k^r(x)$ und analog beschreibt die Funktion $\text{Peak}_k^r(x)$ die Dyck-Pfade mit genau r Peaks auf Höhe k . Damit können wir den Zusammenhang zwischen Peaks und Tälern beschreiben.

Satz 2.27. *Für $k \geq 2$ gilt*

$$\text{Peak}_k^r(x) = \text{Valley}_{k-2}^r(x). \tag{8}$$

Beweis. Wir beweisen den Satz, indem wir eine Bijektion Φ zwischen den Dyck-Pfaden der Länge $2n$ mit genau r Peaks auf Höhe k und den Dyck-Pfaden mit exakt r Tälern auf Höhe $k - 2$ angeben. Dazu betrachten wir einen Dyck-Pfad P als Abfolge der Gitterpunkte P_i , wobei P_i der Koordinate nach dem i -ten Schritt entspricht. $\Phi(P) = \phi(P_1)\phi(P_2) \dots \phi(P_{2n})$, wobei $\phi(P_i) = P_i + (0, 2)$, falls es sich bei P_i um ein Tal auf Höhe $k - 2$ handelt. Es gilt $\phi(P_i) = P_i - (0, 2)$ falls es sich um einen Peak auf Höhe k handelt und $\phi(P_i) = P_i$ sonst.

Φ ist wohldefiniert, da nach einem Peak auf Höhe k ein Punkt auf Höhe $k - 1$ folgt. Analog kommt nach einem Tal mit Höhe $k - 2$ ein Punkt mit Höhe $k - 1$. Mit der Anwendung von Φ erhalten wir also wieder einen Dyck-Pfad, da Peaks auf Höhe k mit Tälern auf Höhe $k - 2$ vertauscht werden. Daraus folgt auch:

2 Gitterpfade

$\Phi = \Phi^{-1}$. Damit ist Φ eine Bijektion.

□

Damit lässt sich etwas unseres Wissens über Peaks auf Täler übertragen.

Korollar 2.28. *Es gilt*

$$\text{Peak}_2^0(x) = \text{Valley}_0^0(x) = 1 + xC(x).$$

Ein Tal auf Höhe 0 kann auch als Berühren der x-Achse interpretiert werden. Die Zahl der n-Dyck-Pfade, welche nur mit dem Start- bzw. Endpunkt die x-Achse berühren, beträgt C_{n-1} . Diese Tatsache kann man auch konstruktiv darstellen, da ein solcher Dyck-Pfad mit einem Up-Step beginnen muss. Darauf folgt ein beliebiger (n-1)-Dyck-Pfad, gefolgt von einem Down-Step.

Beweis. (Satz 2.20)

Sei Φ_k die Funktion aus dem Beweis für Satz 2.27, welche Valleys auf Höhe k mit Peaks auf Höhe $k + 2$ vertauscht. Für einen Dyck-Pfad ohne Peak auf Höhe $k + 2$ wird durch das Anwenden von Φ_k kein Peak auf Höhe $k + 1$ erzeugt. Ein Valley auf Höhe $k + 1$ ist jedoch möglich. Sei nun d ein Dyck-Pfad ohne Peaks in der Menge $\{2, \dots, m\}$ so können wir durch Φ_k einen Dyck-Pfad erzeugen, welcher kein Valley in $\{0, \dots, m - 2\}$ hat, nämlich $\Phi_{m-2} \circ \dots \circ \Phi_0(d)$. Dies ist eine Bijektion, da es sich um eine Zusammensetzung von Bijektionen handelt. Für einen solchen Pfad gibt es zwei Möglichkeiten. Erstens kann er die Form $U^k D^k$ für $0 \leq k \leq m - 2$ haben. Die zweite Möglichkeit ist, dass er von der Bauart $U^{m-1} \tilde{d} D^{m-1}$ ist, wobei \tilde{d} ein beliebiger Dyck-Pfad ist. Damit ergibt sich als erzeugende Funktion

$$\text{Peak}_{\{2, \dots, m\}}(x) = 1 + x + \dots + x^{m-2} + x^{m-1}C(x).$$

□

Wir beenden unseren kleinen Ausflug und kehren nun wieder zum eigentlichen Grund dieses Kapitels zurück. Wir können die erzeugende Funktion $\text{Valley}_k^r(x)$ berechnen und mit Gleichung (8) bekommen wir für $k \geq 2$ die Funktion $\text{Peak}_k^r(x)$. Schlussendlich müssen wir noch die Funktion $\text{Peak}_1^r(x)$ auf einem anderen Weg berechnen.

Folgender Satz benötigt Chebyshev-Polynome zweiter Art.

2 Gitterpfade

Definition 2.29. Die Chebyshev-Polynome zweiter Art sind definiert als

$$U_r(\cos \theta) = \frac{\sin(r+1)\theta}{\sin \theta}.$$

Desweiteren benötigen wir noch die rationalen Polynome $R_k(x)$, welche durch

$$R_k(x) = \frac{U_{k-1}\left(\frac{1}{2\sqrt{x}}\right)}{\sqrt{x}U_k\left(\frac{1}{2\sqrt{x}}\right)}$$

definiert sind.

Satz 2.30. Für alle $r \geq 0$ gilt

$$\text{Peak}_1^r(x) = \delta_{r,0} + \frac{x^{3r+2}C(x)^{2r+2}}{(1-x^2C(x)^2)^{r+1}},$$

und für $k, r \geq 0$ gilt

$$\text{Valley}_k^r(x) = \delta_{r,0}R_{k+1}(x) + \frac{x^r C(x)^{r+1}}{U_{k+1}^2\left(\frac{1}{2\sqrt{x}}\right)(1-x(R_{k+1}(x)-1)C(x))^{r+1}}.$$

Der Beweis dieses Satzes ist recht technisch und wird in [3] ab Seite 4 geführt.

2.5 Klassische Mustervermeidung

Dieses Unterkapitel trägt den Titel "Klassische Mustervermeidung", da der Inhalt des nächsten Abschnitts der Mustervermeidung in Permutationen am nächsten kommt. Wir orientieren uns am Paper [5] und werden wie gehabt Dyck-Pfade behandeln, jedoch werden wir verschiedene Muster bearbeiten, welche die gemeinsame Eigenschaft haben, dass sie nicht zusammenhängend auftreten müssen. Was das genau bedeutet, werden wir in der nächsten Definition sehen.

Definition 2.31. Sei $w = v_1v_2 \dots v_{2n}$ ein beliebiger n -Dyck-Pfad und $p = p_1 \dots p_k$ eine Folge der Länge k aus dem Alphabet $\{U, D\}$. Der Dyck-Pfad w enthält p , genau dann wenn es i_1, \dots, i_k aus den natürlichen Zahlen gibt mit $i_1 \leq i_2 \leq \dots \leq i_k$, sodass für alle $j \leq k$ gilt $v_{i_j} = p_j$. In diesem Fall schreiben wir $p \leq w$. Andernfalls vermeidet w das Muster p .

Die Menge $\mathcal{D}_n(p)$ bezeichnet die Menge aller n -Dyck-Pfade, welche das Muster p vermeiden und $d_n(p)$ beschreibt die Mächtigkeit dieser Menge. Wir werden am folgenden Beispiel das Ganze verdeutlichen.

2 Gitterpfade

Beispiel 2.32. Sei $p = (UD)^k$. Für $k = 1$ ist unser Muster ein einfacher Up-Step, gefolgt von einem Down-Step. Klarerweise muss jeder nicht triviale Dyck-Pfad dieses Muster enthalten. Damit gilt $d_n(p) = 0$ für alle $n \geq 1$.

Sei k beliebig, dann enthält ein Dyck-Pfad genau dann dieses Muster, falls er k oder mehr Peaks hat. Damit muss die Zahl der n -Dyck-Pfade, welche das Muster p vermeiden, gleich der Summe der n -Dyck-Pfade mit $j < k$ Peaks sein. Dank der Bemerkung 2.10 über die Narayana-Zahlen erhalten wir

$$d_n((UD)^k) = \sum_{j=1}^{k-1} N(n, j).$$

Proposition 2.33. Sei p ein beliebiges Muster der Länge $2k$, dann gilt $d_n(p) = C_n$ für $n < k$. Für $n = k$ gilt $d_n(p) = C_n - 1$.

Nun wollen wir uns einem etwas anspruchsvolleren Muster widmen. Für beliebiges k betrachten wir das Muster $p_k = U^{k-1}DUD^{k-1}$.

Satz 2.34. Es gilt

$$d_n(p_k) = \begin{cases} C_{k-1}^2, & \text{für } n \geq 2k - 3 \\ \sum_{j \geq 1} b_{k-j, n-k+j}^2, & \text{sonst.} \end{cases}$$

Beweis. Sei A der Endpunkt des $(k - 1)$ -ten Up-Steps. Dieser muss auf der Geraden $r: y = -x + 2k - 2$ liegen. Analog dazu betrachten wir den Punkt B, welcher den Startpunkt jenes Down-Steps bezeichnet, auf den noch weitere $k - 2$ folgen. Der Punkt B muss auf der Geraden $s: y = x - (2n - 2k + 2)$ liegen. Die Geraden r und s haben klarerweise genau einen Schnittpunkt und wir müssen verschiedene Fälle je nach Lage dieses Schnittpunkts unterscheiden.

1.) Für $2n \geq 2k - 3$ liegt der Schnittpunkt der Geraden r und s auf einer Höhe kleiner gleich 1. Damit muss gelten, dass die x -Koordinate von A oder auch x_A kleiner sein muss als die x -Koordinate von B, x_B . Das bedeutet, wir können einen beliebigen Dyck-Pfad d , welcher das Muster p_k vermeidet, in drei Abschnitte unterteilen. Der erste Abschnitt d_A geht von $(0, 0)$ zum Punkt A, der zweite von A zu B und der dritte Abschnitt d_B geht von B zum Punkt $(0, 2n)$. Für d_A gibt es natürlich gleich viele Kombinationen wie für d_B . Würde man an d_A noch genügend Down-Steps anhängen, so bekäme man jeden beliebigen $(k - 1)$ -Dyck-Pfad. Das bedeutet, es gibt C_{k-1} Kombinationen für d_A . Nun zum Abschnitt zwischen A und B. Bis zum Punkt A gab es bereits $k - 1$ Up-Steps und nach dem Punkt B wird es noch $k - 1$ Down-Steps geben. Damit darf das Muster DU nicht in der Verbindung von A und B vorkommen. Es gibt damit nur mehr eine Möglichkeit, $U^i D^j$. Der dritte Abschnitt funktioniert analog zum

2 Gitterpfade

ersten. Dieser Fall hat damit C_{k-1}^2 Kombinationen.

2.) Die Geraden r und s treffen sich auf einer Höhe größer 1. In diesem Fall gibt es wieder zwei Möglichkeiten, je nach dem wie die x -Koordinaten von A und B zueinander liegen.

a) $x_A \leq x_B$: In diesem Fall verwenden wir ein sehr ähnliches Argument zum obigen. Der Dyck-Pfad wird wieder in drei Teile unterteilt, wobei für den zweiten nur eine Möglichkeit frei bleibt, nämlich $U^i D^j$. Allerdings gibt es für den Abschnitt eins und drei eine zusätzliche Restriktion. Wie oben werden wir nur Abschnitt eins behandeln. Die Aussage $x_A \leq x_B$ entspricht der Aussage, dass x_A zwischen den Werten $k - 1$ und n liegt. Die einzelnen Möglichkeiten sind daher beschränkt auf Wege mit $k - 1$ Up-Steps und 0 bis $n - k + 1$ Down-Steps, wobei natürlich vor dem i -ten Down-Step schon i Up-Steps gemacht worden sein müssen. Sei $b_{i,j}$ die Zahl ¹ dieser Wege, dann gilt für diesen Fall

$$d_n^{(a)}(p_k) = \left(\sum_{j=0}^{n-k+1} b_{k-2,j} \right)^2. \quad (9)$$

b) $x_A > x_B$: In diesem Fall wird das Muster offensichtlich vermieden, da nachdem die $k - 1$ Up-Steps gemacht wurden, gar nicht mehr genug Down-Steps kommen, um das Muster zu erzeugen. Wir unterteilen hier den Pfad in zwei Abschnitte, welche sich am Punkt C treffen. Dieser Punkt C hat die x -Koordinate n und seine y -Koordinate ist größer gleich 0 und echt kleiner als der Schnittpunkt der beiden Geraden. Die Argumentation für die beiden Fälle läuft wieder analog. Bis zum Punkt C können höchstens $k - 2$ Up-Steps gemacht worden sein, der Rest müssen Down-Steps sein. Damit gilt

$$d_n^{(b)}(p_k) = \left(\sum_{j=2}^k b_{k-j,n-k+j} \right)^2. \quad (10)$$

Aus (9) und (10) erhalten wir

$$d_n(p_k) = d_n^{(a)}(p_k) + d_n^{(b)}(p_k) = \left(\sum_{j=0}^{n-k+1} b_{k-2,j} \right)^2 + \left(\sum_{j=2}^k b_{k-j,n-k+j} \right)^2.$$

Nach Proposition 2.33 müssen wir nur die Fälle $n \geq k + 1$ betrachten. Unter dieser Bedingung ist $d_n^{(a)}(p_k)$ nicht null und nach Gleichung (48) können wir den Term zu dem Gewünschten vereinfachen.

□

¹Die Zahlen $b_{i,j}$ entsprechen den Ballot-Zahlen siehe Appendix 6.3

2 Gitterpfade

Nun ein zweites sehr ähnliches Muster.

Satz 2.35. *Sei \tilde{p}_k das Muster $U^k D^k$. Dann gilt*

$$d_n(\tilde{p}_k) = \begin{cases} 0, & \text{für } n \geq 2k - 1 \\ \sum_{j \geq 1} b_{k-j, n-k+j}^2, & \text{sonst.} \end{cases}$$

Beweis. Das Muster \tilde{p}_k kann dargestellt werden als $U^{k-1} U D D^{k-1}$. Der Beweis verläuft analog zum obigen. Wir unterteilen die Dyck-Pfade in drei Abschnitte, wobei der zweite Abschnitt diesmal mit einer Ahnzahl an Down-Steps beginnt und danach kommen Up-Steps. Für den Fall $n \geq 2k - 1$ müssten jedoch so viele aufeinanderfolgende Down-Steps gemacht werden, dass die x-Achse unterschritten würde. Damit ist in diesem Fall 0 das Ergebnis. Fall zwei verläuft analog zum obigen. □

Wir wollen dieses Unterkapitel mit einer Vermutung beenden [5, S. 693].

Vermutung 2.36. ² *Sei P ein n -Dyck-Pfad welcher mit a Up-Steps beginnt und mit b Down-Steps endet. Sei $k = 2n - 2 - a - b$, dann ist $d_n(P)$ asymptotisch zu*

$$\frac{\alpha_P \cdot C_a \cdot C_b}{k!} n^k, \tag{11}$$

wobei α_P die Zahl der maximalen Ketten im Dyck-Verband der Ordnung n von P zum maximalen Element $U^n D^n$.

2.6 Mustervermeidung in Gitterpfaden

Nun kommen wir zum wahrscheinlich wichtigsten Teil dieses Kapitels, der Mustervermeidung in Gitterpfaden. Wir werden beliebige, aber zusammenhängende Muster betrachten und wir werden allgemeine Gitterpfade statt Dyck- oder Motzkin-Pfaden betrachten. Die Inhalte stammen aus [6] und [7].

Definition 2.37. Sei S Teilmenge von \mathbb{Z} , mit mindestens einer positiven und einer negativen Zahl. Ein Gitterpfad w mit Schritten aus S ist definiert als endliche Folge $w = v_1 v_2 \dots v_n$, wobei v_i in der Menge S enthalten sein muss. Die Länge von w oder auch $|w|$ entspricht der Zahl der Schritte in w und $h(w) = v_1 + v_2 + \dots + v_n$ beschreibt die abschließende Höhe des Pfades. Ein solcher Gitterpfad wird auch Weg genannt.

²Siehe [5] Seite 693

2 Gitterpfade

Wie man sieht, ist die Definition eines Wegs um einiges allgemeiner als die der Dyck-Pfade. Ein solcher Gitterpfad kann genau so als String über dem Alphabet S interpretiert werden. Mehr dazu folgt im Kapitel über Strings.

Definition 2.38. Ein Weg, dessen Endpunkt sich auf der x-Achse befindet, wird auch Brücke genannt.

Für eine Brücke b muss natürlich $h(b) = 0$ gelten und klarerweise muss es nicht immer Brücken der Länge n geben, wie das folgende Beispiel zeigt.

Beispiel 2.39. Sei S die Menge $\{-1, 1\}$, dann ist die Zahl der Brücken der Länge n gleich 0 für ungerade Zahlen. Für die geraden Zahlen erhalten wir das Ergebnis aus dem Binomialkoeffizienten. Es muss $k = n/2$ mal der Schritt -1 vorkommen und genau so oft auch der Schritt $+1$. Damit kommen wir auf die Lösung

$$|\{b : b \text{ Brücke der Länge } n\}| = \begin{cases} \binom{2k}{k}, & \text{falls } n \text{ gerade} \\ 0, & \text{sonst.} \end{cases}$$

Definition 2.40. Ein Weg, dessen Endpunkt sich nicht auf der x-Achse befindet, aber nie die x-Achse unterschreitet, wird Mäander genannt. Wir sprechen von einem Pseudomeander, falls der Weg, ausgenommen von seinem letzten Schritt, nie unter die x-Achse fällt. Der letzte Schritt darf, muss aber nicht die x-Achse unterschreiten.

Definition 2.41. Eine Brücke, welche nie die x-Achse unterschreitet, wird als Exkursion bezeichnet.

Bemerkung 2.42. Sei $S = \{-1, 1\}$, dann entsprechen die Exkursionen mit Schritten aus S den Dyck-Pfaden. Die Motzkin-Pfade sind Exkursionen mit Schritten aus $\{-1, 0, 1\}$.

Unser Ziel ist es die erzeugende Funktion für die verschiedenen Gitterpfade, welche ein zusammenhängendes Muster vermeiden, herzuleiten. Die dafür notwendige Theorie werden wir uns jetzt erarbeiten.

Definition 2.43. Ein Gitterpfad $w = v_1 v_2 \dots v_n$ über S enthält ein zusammenhängendes Muster $p = p_1 \dots p_k$ mit $p_i \in S$, genau dann, wenn es einen Index $1 \leq j \leq n$ gibt, sodass $v_j = p_1, v_{j+1} = p_2 \dots$ und $v_{j+k-1} = p_k$ gilt.

Diese Definition ist klarerweise restriktiver als Definition 2.31. Ein Gitterpfad welcher ein Muster nach Definition 2.43 enthält, tut dies auch nach Definition 2.31. Jedoch gilt dies nicht umgekehrt.

2 Gitterpfade

Definition 2.44. Sei S eine Teilmenge von \mathbb{Z} mit mindestens einer positiven und einer negativen Zahl. Sei $-c$ die kleinste Negative und d die größte positive Zahl, dann ist

$$P(u) = u^{-c} + \dots + u^d \quad (12)$$

das so genannte charakteristische Polynom.

Als nächstes beschäftigen wir uns mit dem Thema Autokorrelation. Sei S ein beliebiges Alphabet. Ein Präfix eines Wortes $w = w_1 \dots w_n$ ist ein Wort $p = p_1 \dots p_k$, wobei $w_i = p_i$ für alle i von 1 bis k gelten muss. Klarerweise ist das leere Wort ϵ und das ganze Wort w ein Präfix von w . Der Ausdruck \bar{p} bezeichnet das Komplement von p in w , also das Wort $\bar{p} = w_{k+1} \dots w_n$. Ein Suffix ist das Analogon zu einem Präfix, welches sich auf den Abschluss eines Wortes bezieht.

Definition 2.45. Ein Präsuffix eines Wortes w ist ein Wort, welches sowohl Präfix als auch Suffix von w ist.

Beispiel 2.46. Sei w das Wort "MAMA", dann ist die Menge der Präsuffixe Q_w des Wortes w gleich $\{\epsilon, MA, MAMA\}$.

Wenn ein Wort nicht nur die beiden trivialen Präsuffixe, das leere Wort und das Wort selbst zulässt, so sprechen wir von Autokorrelation.

Definition 2.47. Sei w ein Wort, Q_w^+ die Menge der nichtleeren Präsuffixe. Dann wird mit

$$R(t, u) := \sum_{q \in Q_w^+} t^{|\bar{q}|} u^{h(\bar{q})} \quad (13)$$

das so genannte Autokorrelationspolynom von w bezeichnet.

Beispiel 2.48. Sei w das Wort "10101", dann ist Q_w^+ gleich der Menge $\{1, 101, 10101\}$. Für das zugehörige Autokorrelationspolynom gilt

$$R(t, u) = 1 + t^2 u^{h(0+1)} + t^4 u^{0+1+0+1}.$$

Nun kommen wir zu unserer letzten Definition.

Definition 2.49. Der Kern eines Gitterpfades p über der Schrittmenge S , ist das Laurent-Polynom

$$K(t, u) := (1 - tP(u))R(t, u) + t^{|p|} u^{h(p)}. \quad (14)$$

2 Gitterpfade

Nach [6] kann gezeigt werden, dass für jede Lösung mit $u = u(t)$ von $K(t, u) = 0$ entweder $\lim_{t \rightarrow 0} u(t) = 0$ - in diesem Fall sprechen wir von einer kleinen Lösung - oder $\lim_{t \rightarrow 0} u(t) = \infty$ gilt - hier sprechen wir von einer großen Lösung. Die Zahl der kleinen Lösungen wird mit $e := \max\{c, -h(p)\}$ bezeichnet, und die einzelnen Lösungen werden mit u_1, \dots, u_e bezeichnet. Nun folgt einer der wohl wichtigsten Sätze dieser Arbeit. Die Variable t beschreibt die Länge der Pfade und u die abschließende Höhe.

Satz 2.50. *Sei S eine Menge an Schritten, p ein beliebiges Muster aus Schritten aus S , dann entspricht die bivariate erzeugende Funktion*

$$W(t, u) = \frac{R(t, u)}{K(t, u)} \quad (15)$$

der erzeugenden Funktion der Wege, welche das Muster p vermeiden.

Um diesen Satz beweisen zu können, benötigen wir noch Wissen über endliche Automatentheorie.

Definition 2.51. Ein endlicher Automat \mathcal{A} ist ein 5 - Tupel $\langle X, X_0, S, Y, \tilde{A} \rangle$, wobei X die Menge an Zuständen bezeichnet, in denen sich der Automat befinden kann. Die zweite Komponente, X_0 , bezeichnet den Startzustand und muss Teil der Zustandsmenge X sein. Durch S wird das akzeptierte Alphabet beschrieben und durch die vierte Komponente wird die Endzustandsmenge, welche wieder Teil der gesamten Zustandsmenge sein muss, definiert. Als letzter Teil kommt \tilde{A} , die Übergangsfunktion des Automaten. Diese beschreibt, falls sich der Automat in Zustand x befindet und Eingabe s erfolgt, welcher der nächste Zustand ist.

$$\begin{aligned} \tilde{A} : X \times S &\rightarrow X \\ (x, s) &\rightarrow \tilde{A}(x, s) \end{aligned}$$

Was bedeutet das jetzt für unser konkretes Beispiel?

Für ein vorgegebenes Alphabet S und Muster $p = p_1 \dots p_n$ über S benötigen wir einen Automaten \mathcal{A} . Wir werden unseren Pfad Schritt für Schritt konstruieren und \mathcal{A} zeichnet auf, wie viel vom Muster generiert wurde. Das bedeutet, für ein Muster der Länge n benötigen wir n verschiedene Zustände $X = \{X_0, \dots, X_{n-1}\}$. Der Startzustand ist X_0 , welcher repräsentiert, dass noch kein Teil des Musters erzeugt wurde. Die Zustände X_i benötigen wir um anzuzeigen, dass das Muster bis zum i -ten Schritt aufgebaut wurde. Da wir nie das ganze Muster erzeugen, benötigen wir keinen Zustand X_n und jeder der n Zustände ist als Endzustand möglich. Jetzt müssen wir nur noch die Übergangsfunktion \tilde{A} beschreiben. Klarerweise kommen wir von X_{i-1} und der Eingabe p_i in den nächsten Zustand X_i .

2 Gitterpfade

Da das Muster nur Schritt für Schritt aufgebaut werden kann, ist es nur möglich von X_{i-1} nach X_i zu springen, nicht jedoch weiter. Falls im Zustand X_{i-1} eine andere Eingabe $\lambda \neq p_i$ erfolgt, so waren die letzten Eingaben $w = p_1 p_2 \dots p_{i-1} \lambda$. Sei q der längste Suffix von w , welcher auch Präfix von p ist. Damit haben wir bereits $|q|$ viele Teile von p erzeugt und damit muss gelten, dass der auf X_{i-1} folgende Zustand mit Eingabe λ gleich $X_{|q|}$ sein muss, also $\tilde{A}(X_{i-1}, \lambda) = X_{|q|}$. Damit hat unser Automat die Form

$$\mathcal{A} = \langle X, X_0, S, X, \tilde{A} \rangle. \quad (16)$$

Für unsere Berechnungen benötigen wir jedoch die Transfermatrix A . Deren Einträge a_{ij} sind definiert als Summe aller u^λ mit $\tilde{A}(X_i, \lambda) = X_j$.

Definition 2.52. Für eine gegebene Matrix B ist die Adjunkte $\text{adj}(B)$ definiert als die Matrix mit den Einträgen $b_{ij} = (-1)^{i+j} M_{ji}$. Die Einträge M_{ji} entsprechen den Determinanten jener Matrizen, welche man durch streichen der j -ten Zeile und i -ten Spalte aus der Matrix B erhält.

Satz 2.53. Sei S die Menge der Schritte, p ein Muster über S und A die Transfermatrix des zugehörigen Automaten. Dann gilt

$$\det(I - tA) = K(t, u) = (1 - tP(u))R(t, u) + t^{|p|} u^{h(p)}, \quad (17)$$

$$(1 \ 0 \ \dots \ 0) \text{adj}(I - tA) (1 \ 1 \ \dots \ 1)^T = R(t, u). \quad (18)$$

Eine Beweisidee dieses Satzes gibt es in [6] auf Seite 7.

Beweis. (Satz 15)

Sei $p = p_1 \dots p_n$ ein Muster, $W(t, u)$ die erzeugende Funktion der Wege welche das Muster p vermeiden und $k \in \{0, \dots, n-1\}$, $W_k = W_k(t, u)$ die erzeugende Funktion der Wege welche p vermeiden und das Muster bis zum k -ten Schritt aufgebaut haben.

Wir versuchen nun eine Funktionalgleichung für den Vektor $(W_0 \dots W_{n-1})$ niederzuschreiben. Klarerweise kann der triviale Weg als $(1 \ 0 \ \dots \ 0)$ interpretiert werden. Für einen nicht trivialen Weg w der Länge n gilt, dass man verschiedene Wege der Länge $(n+1)$ erhalten kann, falls man einen zusätzlichen Schritt hinzufügt. Wir haben die Transfermatrix A unseres Automaten \mathcal{A} so konstruiert, dass sie genau hierfür verwendet werden kann.

$$(W_0 \dots W_{n-1}) = (1 \ 0 \ \dots \ 0) + t(W_0 \dots W_{n-1})A$$

Aus dieser Gleichung und durch einfache Umformungen erhalten wir

2 Gitterpfade

$$\begin{aligned} (W_0 \dots W_{n-1})(I - tA) &= (10 \dots 0), \\ (W_0 \dots W_{n-1}) &= (10 \dots 0) \frac{\text{adj}(I - tA)}{\det(I - tA)}. \end{aligned}$$

Die Funktion $W(t, u)$ entspricht klarerweise der Summe der W_k , was auch als Multiplikation mit dem Vektor $(1 \dots 1)$ interpretiert werden kann. Damit und aus dem obigen Satz ergibt sich die Formel

$$W(t, u) = \frac{R(t, u)}{K(t, u)}.$$

□

Bemerkung 2.54. Um die erzeugende Funktion der Brücken $B(t)$ zu erhalten, müssen wir die Wege betrachten, welche auf der x-Achse terminieren, also jene Terme mit dem Koeffizienten u^0 . Es gilt also

$$B(t) = [u^0]W(t, u).$$

Mithilfe der Cauchy'schen Integralformel und dem Residuensatz erhalten wir

$$[u^0]W(t, u) = \frac{1}{2\pi i} \int_{|u|=\epsilon} \frac{W(t, u)}{u} du = \sum_{i=1}^e \text{Res}_{u=u_i(t)} \frac{W(t, u)}{u},$$

wobei $u_1(t), \dots, u_e(t)$ die kleinen Lösungen von $K(t, u)$ bezeichnet. Durch einige Rechenschritte wie in [6] ergibt sich schlussendlich die Formel

$$B(t) = \sum_{i=1}^e \frac{u_i'}{u_i} \frac{t}{1 + \frac{R'(t, u_i)}{R(t, u_i)^2} t^{n+1} u_i^{h(p)} - \frac{(l-1)t^n u_i^{h(p)}}{R(t, u_i)}}. \quad (19)$$

Für Mäander und Exkursionen gibt es etwas ästhetischere Terme.

Satz 2.55. *Sei S eine Menge von Schritten und $p = p_1 \dots p_n$ ein Pseudomeander, dann lässt sich die erzeugende Funktion $M(t, u)$ der Mäander welche das Muster p vermeiden, darstellen als*

$$M(t, u) = \frac{R(t, u)}{u^c K(t, u)} \prod_{i=1}^c (u - u_i(t)), \quad (20)$$

2 Gitterpfade

wobei $u_1(t), \dots, u_c(t)$ die kleinen Lösungen von $K(t, u) = 0$ sind. Die erzeugende Funktion für Exkursionen $E(t)$ bekommt man aus der obigen Gleichung, denn es gilt

$$E(t) = M(t, 0) = \frac{(-1)^{c+1}}{t} \prod_{i=1}^c u_i(t). \quad (21)$$

Für den Fall $h(p) = -c$ steht im Zähler der Term $t - t^n$ an Stelle von t .

Beweis. Der Anfang dieses Beweises funktioniert analog zum obigen. Es bezeichnet $M_k = M_k(t, u)$ die erzeugende Funktion der Mäander, welche das Muster p bis zum k -ten Schritt generiert haben. Damit können wir eine sehr ähnliche Gleichung aufstellen, jedoch müssen wir, da die x -Achse nicht unterschritten werden darf, noch einen Term hinzufügen. Dadurch erhalten wir

$$(M_0 \dots M_{n-1}) = (10 \dots 0) + t(M_0 \dots M_{n-1})A - t[u^{<0}]((M_0 \dots M_{n-1})A),$$

wobei $[u^{<0}]$ jene Terme bezeichnet, in denen u mit einem negativen Exponenten auftritt. Ein weiteres wichtiges Argument liefert die Tatsache, dass es sich bei unserem Muster um einen Pseudomeander handelt. Demnach kann während das Muster akkumuliert wird die x -Achse nicht unterschritten werden. Der Term $[u^{<0}]$ können wir also auch als $F(t, u)/u^c$ aufschreiben, wobei F ein Polynom in den Variablen t und u ist. Durch Umformungen erhalten wir

$$(M_0 \dots M_{n-1})(I - tA) = \left(1 - \frac{t}{u^c} F(t, u)\right)(10 \dots 0).$$

Woraus sich analog zum obigen Beweis und durch anschließendes Aufsummieren

$$M(t, u) = \left(1 - \frac{t}{u^c} F(t, u)\right) \frac{R(t, u)}{K(t, u)} \quad (22)$$

ergibt. Der Rest dieses Beweises beschäftigt sich mit der Vereinfachung dieses Ausdrucks und ist in [6] zu finden. □

Nun da wir die großen Sätze dieses Kapitels beendet haben, kommen wir noch zu einer Anwendung.

3 Bäume

Wie der Name dieses Kapitels schon verrät, werden wir im Folgenden verschiedenste Arten von Bäumen bearbeiten. Bäume an sich sind eine sehr reichhaltige Struktur mit vielen Anwendungen, dadurch gibt es aber auch sehr viele Klassen und Familien. Aufgrund der begrenzten Möglichkeiten werden jedoch nur einzelne, diese dafür detaillierter, beschrieben. Das Kapitel ist in zwei Teile unterteilt. Im ersten bearbeiten wir markierte Objekte, wobei exponentiell erzeugende Funktionen vorkommen werden. Im zweiten Teil werden nicht markierte Objekte behandelt, welche durch die gewöhnlichen erzeugenden Funktionen beschrieben werden.

3.1 Heaps

Unsere erste Struktur, mit der wir uns eingehend auseinandersetzen werden, sind die Heaps. Aufgrund der Tatsache, dass es sich hierbei um markierte Objekte handelt, gibt es hier einen sehr einfachen Zugang zur Mustervermeidung in den Permutationen. Wir werden jedem Heap eine Permutation zuordnen und können dann in einer vertrauten Umgebung arbeiten.

Definition 3.1. Ein k -Heap ist definiert als ebener, vollständiger k -Wurzelbaum, wobei jeder Knoten mit einer Zahl aus der Menge $\{1, \dots, n\}$ markiert ist. Jedoch muss die Markierung jedes Kindknotens größer als die des Elternknotens sein. Die Menge der k -Heaps mit n Knoten wird mit \mathcal{H}_n^k bezeichnet.

Wie ein solcher k -Heap mit n Knoten genau aussieht wollen wir nun genauer beschreiben. Es existiert ein ausgezeichnete Knoten, die Wurzel. Jeder Knoten hat k oder weniger Nachfolger, wobei wir bei ebenen Bäumen die Reihenfolge der Nachfolgeknoten beachten. Vollständig bedeutet, dass die Knoten auf allen Ebenen des Baumes bis auf der Letzten k Kindknoten besitzen und in der letzten Ebene sind alle Knoten auf der linken Seite angeordnet.

Definition 3.2. Für einen Heap $H \in \mathcal{H}_n^k$ bezeichnet π_H die mit H assoziierte Permutation. Diese entspricht der Reihenfolge der Markierungen, welche man bei der Breitensuche sukzessive antrifft.

Beispiel 3.3. Wir betrachten einen Heap aus \mathcal{H}_{16}^3 .

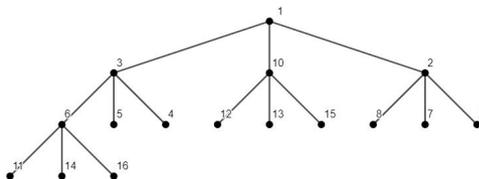


Abbildung 2

3.1.1 Binäre Strauchwälder

Wir werden Heaps noch ganz allgemein betrachten. Der Einfachheit halber beginnen wir jedoch mit den binären Strauchwäldern und den Inhalten aus [8]. Ein binärer Strauch ist ein Binärbaum, wobei die Wurzel nur Blätter als Kinder hat. Da diese Einschränkung sehr restriktiv ist, werden wir Wälder solcher binären Sträucher betrachten. Sei W ein Wald aus binären Sträuchern S_1, \dots, S_n , so bezeichnet $|W|$ die Zahl der Knoten in W , welche Markierungen aus Zahlen aus $\{1, \dots, |W|\}$ tragen. Die mit W assoziierte Permutation π_W ist definiert durch

$$\pi_W := \pi_{S_1} \pi_{S_2} \dots \pi_{S_n}.$$

Die Menge aller k -Strauchwälder mit n Heaps wird mit \mathcal{F}_n^k bezeichnet. Für jeden solchen Wald W kennen wir die assoziierte Permutation π_W und damit können wir mit der Mustervermeidung beginnen. Sei p eine Permutation, so enthält W das Muster p , falls π_W die Permutation enthält, ansonsten vermeidet W das Muster p . Für eine Menge an Permutationen P bezeichnen wir mit $\mathcal{F}_n^k(P)$ die Menge aller k -Strauchwälder mit n Heaps, welche alle Permutationen in P vermeiden. Die Mächtigkeit dieser Menge wird mit $f_n^k(P)$ bezeichnet.

Zunächst werden wir uns auf einzelne Permutationen der Länge drei beschränken, werden dann aber zu Mengen von Permutationen übergehen. Den nächsten Satz über die Vermeidung des Musters $\text{id}_3 = 123$ können wir sogar allgemein für k -Sträucher formulieren.

Satz 3.4. *Für $k \geq 2$ und $n \geq 0$ gilt*

$$f_n^k(123) = \frac{1}{kn + 1} \binom{(k+1)n}{n}.$$

Beweis. Zu Beginn müssen wir festhalten, dass die Markierungen der Wurzelknoten in absteigender Reihenfolge auftreten müssen. Gäbe es zwei Wurzelknoten mit aufsteigender Markierung $w_1 < w_2$, dann würden w_1, w_2 und ein Nachfolger von w_2 eine nicht erlaubte Teilfolge bilden. Aus einem sehr ähnlichen Argument dürfen auch die Nachfolgeknoten nur in absteigender Reihenfolge auftreten. Teil 2 dieses Beweises beschäftigt sich mit der Zahl der Möglichkeiten, die Markierungen so auszuwählen, dass beide Bedingungen erfüllt sind. Wir müssen aus den insgesamt $(k+1)n$ Markierungen n auswählen. Diese bezeichnen wir mit w_i wobei o.B.d.A. $w_1 < w_2 < \dots < w_n$ gilt. Zusätzlich müssen noch Nebenbedingungen gelten, denn nach dem größten Element w_n müssen noch k weitere Markierungen für die Kindknoten frei sein. Desweiteren müssen nach dem w_i

3 Bäume

noch $(n + 1 - i)k$ Markierungen frei sein. Nach Bemerkung 6.13 und einer kurzen Überlegung entspricht dies genau der Zahl der Nord-Ost-Pfade von $(0, 0)$ nach (n, nk) . Damit und mit der Tatsache, dass die ausgewählten Markierungen nur auf eine einzige Art angeordnet werden können, kommen wir auf das gewünschte Ergebnis.

□

Für unsere nächste Aussage benötigen wir einmal mehr Gitterpfade als Hilfsmittel. Insbesondere werden die Schröder-Zahlen S_n benötigt, welche in 6.3 angeführt werden.

Satz 3.5. *Sei S_n die Zahl der Exkursionen von $(0, 0)$ nach $(4n, 0)$ mit Schritten aus $S = \{(1, 3), (2, 2), (1, -1)\}$, dann gilt $S_n = f_n^2(213)$.*

Beweis. Wir zeigen den Satz, indem wir jeder Exkursion mit Schritten aus S eine Permutation zuordnen, welche das Muster 213 vermeidet. Danach zeigen wir, dass es einen binären Strauch Wald mit genau dieser Permutation gibt. Anschließend zeigen wir die umgekehrte Richtung.

Für einen beliebigen Pfad markieren wir jeweils die Schritte $(1, 3)$ und $(2, 2)$. Wobei jeder dieser Schritte drei Markierungen aus den Zahlen $\{1, \dots, 3n\}$ bekommt. Die Start-, Mittel- und Endpunkte der Schritte bekommen Markierungen in folgender Weise. Wir suchen noch unmarkierte Punkte, welche nicht die Mittelpunkte eines $(2, 2)$ Schritts sind, welche die geringste y-Koordinate haben. Gibt es mehrere solche Punkte, so starten wir bei dem mit der größten x-Koordinate. Dieser Punkt bekommt die kleinste noch nicht vergebene Markierung. Anschließend wird rekursiv zuerst der rechte Teilpfad markiert und danach der linke. Sollte ein Teilpfad nur aus einem $(2, 2)$ - Mittelpunkt bestehen, so bekommt dieser Punkt die kleinste noch nicht vergebene Markierung. Nun bekommen wir die Permutation durch Ablesen der Markierungen von links nach rechts. Nach Konstruktion besitzt jeder $(1, 3)$ - Step ein zu 123 äquivalentes Muster, die $(2, 2)$ - Steps ein zu 132 äquivalentes. Aus diesem Grund können wir die erzeugte Permutation einem binären Strauch Wald zuordnen. Zusätzlich wird das Muster 213 vermieden, da für jede Position links davon größere Zahlen stehen als rechts.

Für die zweite Richtung betrachten wir einen binären Strauchwald W und die dazu assoziierte Permutation π_W . Wir erzeugen einen Mäander, indem an den Punkt $(0, 0)$ Schritte anhängen. Wir betrachten der Reihe nach die einzelnen Heaps. Für einen Heap mit aufsteigenden Markierungen der Blätter hängen wir einen $(1, 3)$ Schritt an, ansonsten einen $(2, 2)$ Schritt. Danach müssen die $(1, 1)$ Schritte an der richtigen Position eingefügt werden. Innerhalb der Permutation π_W werden hierfür Stellen markiert. Für zwei aufeinanderfolgende Blätter mit

3 Bäume

fallenden Markierungen wird das erste markiert. Danach wird für jede Wurzel die Zahl der noch nicht markierten Zahlen, welche eine größere Markierung haben, gezählt. Dies ist die Zahl der $(1, -1)$ Steps, welche direkt vor dem zur Wurzel korrespondierenden Schritt eingefügt werden. Nun werden die gerade genutzten Zahlen markiert und der Vorgang wird wiederholt. Schlussendlich werden noch so viele $(1, -1)$ Steps hinten angefügt, dass die x-Achse erreicht wird.

□

Damit kommen wir sofort zu unserem nächsten Ergebnis, denn auch für Strauchwälder, welche das Muster 312 vermeiden, gibt es eine Bijektion zu den oben genannten Exkursionen.

Satz 3.6. *Es gilt*

$$f_n^2(213) = f_n^2(312) = S_n.$$

Der Beweis hierfür würde den Rahmen dieser Arbeit sprengen und wird deshalb hier nicht geführt. Er befindet sich in [8] auf 9,10 und 11.

3.1.2 Binäre Heaps

In diesem neuen Unterkapitel geht es weiterhin um Heaps und wir verwenden die Inhalte aus [10]. Die Strauch Eigenschaft wird aufgehoben und anstatt von Wäldern betrachten wir nur noch einzelne Bäume. Wie üblich beginnen wir mit etwas simpleren Mustern und arbeiten uns Schritt für Schritt zu komplexeren Gebieten vor.

Wie üblich ist es das Ziel die Zahl der Objekte zu berechnen, welche ein Muster in der Menge P vermeiden. Als Notation verwenden wir hierfür $h_n(P) = |\mathcal{H}_n^2(P)|$.

Satz 3.7. *Sei $n \geq 1$, dann gilt*

$$h_n(123) = \begin{cases} 1, & \text{falls } n \leq 3 \\ 0, & \text{sonst.} \end{cases}$$

Beweis. Es gibt genau einen Heap mit einem beziehungsweise zwei Knoten. Diese vermeiden klarerweise das Muster. Von den zwei Heaps mit drei Knoten vermeidet der eine das Muster, der andere nicht. Für $n \geq 4$ muss jeder Heap das Muster enthalten, da er mindestens drei Ebenen hat. Aus der Art und Weise wie Heaps markiert werden folgt, dass das Muster enthalten wird.

□

Auch das Muster 132 ist noch recht simpel.

3 Bäume

Satz 3.8. Für alle $n \geq 1$ gilt

$$h_n(132) = 1$$

Beweis. Da die Wurzel mit der 1 markiert sein muss, ist es, um das Muster 132 zu vermeiden, notwendig, dass der Rest des Heaps in aufsteigender Reihenfolge markiert ist. Es gibt genau einen Heap, welcher diese Bedingung erfüllt. \square

Wir setzen mit Sätzen über die Muster 213 und 231 fort. Hier treten die Catalan-Zahlen auf, was das Ganze etwas interessanter gestaltet.

Satz 3.9. Sei $n \geq 1$, dann gilt

$$h_n(213) = C_{\lceil \frac{n}{2} \rceil}.$$

Beweis. Ein Heap, welcher das Muster 213 vermeidet, darf keine internen Knoten haben, deren Markierungen in absteigender Reihenfolge auftreten, da diese das Muster 21 bilden würden. Zusammen mit dem Kindknoten würde dann ein Widerspruch auftreten. Desweiteren müssen interne Knoten direkt aufeinander folgende Markierungen haben, denn angenommen für Zahlen $a < b < c < d$ zwei interne Knoten würden die Markierungen a und c tragen. Weiters sei d die Markierung eines Kindknotens von c . Dann muss gelten, dass die Markierung b vor d auftritt, da jeder Knoten nach c eine größere Markierung als c trägt. Wäre b dessen Kindknoten, so wäre das ein Widerspruch dazu, wie man Heaps markiert. Damit müsste aber das Muster $acbd$ auftreten, wobei cbd dem Muster 213 entspricht - ein Widerspruch. Wir wissen also, dass es nur eine eindeutige Möglichkeit gibt, die internen Knoten zu markieren. Die Blätter dürfen beliebig markiert werden, ohne jedoch die Permutation 213 zu enthalten. Nach Beispiel 1.5 werden diese Permutationen durch die Catalan-Zahlen gezählt. Damit und mit der Tatsache, dass der Heap genau $\lceil \frac{n}{2} \rceil$ Blätter hat, folgt das Gewünschte. \square

Satz 3.10. Es gilt $h_1(231) = 1$ und für $n \geq 2$ gilt

$$h_n(231) = \sum_{i=0}^{\lceil n/2 \rceil - 1} C_i \cdot h_{n-i-1}^2(231).$$

Beweis. Sei H ein Heap aus $\mathcal{H}_n^2(231)$. Wir betrachten die Position der Markierung n , welche auf einem der Blätter auftreten muss. Damit kann die zu H assoziierte Permutation π_H dargestellt werden als $\pi_1 \pi_2 \dots \pi_{n-i-1} n \pi_{n-i+1} \dots \pi_n$, wobei noch $0 \leq i \leq \lceil \frac{n}{2} \rceil - 1$ Markierungen nach n auftreten. Wir wissen, dass π_H das Muster 231 vermeiden muss. Daher müssen die Markierungen $\pi_1 \pi_2 \dots \pi_{n-i-1}$

3 Bäume

kleiner als $\pi_{n-i+1} \dots \pi_n$ sein. Die i Markierungen der Blätter rechts von n müssen die Permutation 231 vermeiden und es gibt C_i viele Möglichkeiten, diese anzuordnen. Die Elemente $\pi_1 \pi_2 \dots \pi_{n-i-1}$ müssen wieder einen Heap bilden, welcher das Muster 231 vermeidet. Dafür gibt es $h_{n-i-1}(231)$ Möglichkeiten. Das gewünschte Ergebnis bekommt man über das Aufsummieren über alle i zwischen 0 und der Zahl der Blätter minus 1.

□

Von besonderem Interesse ist der nächste Satz, da wir unsere erste Wilf-Äquivalenz für Heaps zeigen können.

Satz 3.11. *Für $n \geq 1$ gilt*

$$h_n(231) = h_n(312).$$

Der Beweis für diesen Satz ist recht simpel aber langwierig. Es existiert eine Bijektion zwischen beiden Strukturen, welche in [10] zu finden ist.

Bemerkung 3.12. Das Abzählproblem des Musters 321 ist derzeit noch offen.

Nun, da wir Aussagen über alle Muster der Länge drei getroffen haben, setzen wir mit Mengen von Mustern der Länge drei fort. Bemerkenswert ist, dass die Wilf-Äquivalenz der Muster 231 und 312 erhalten bleibt, wenn wir zusätzlich das Muster 213 vermeiden.

Satz 3.13. *Für $n \geq 1$ gilt*

$$h_n(213, 231) = 2^{\lceil \frac{n}{2} \rceil - 1}.$$

Beweis. Der Beweis ist klarerweise recht ähnlich zu dem von Satz 3.10. Wir wissen auch, dass die größte Markierung n auf einem der Blätter auftreten muss. Links davon stehen kleinere Elemente als rechts. Da wir das Muster 213 vermeiden, sind die Markierungen der internen Knoten in aufsteigender Reihenfolge sortiert, beginnend mit 1 bei der Wurzel. Für die $\lceil \frac{n}{2} \rceil$ Blätter gilt, dass sie eine Permutation bilden müssen, welche sowohl 231 als auch 213 vermeidet. Nach Beispiel 1.6 erhalten wir das gewünschte Ergebnis, $2^{\lceil \frac{n}{2} \rceil - 1}$.

□

Satz 3.14. *Sei $n \geq 1$ dann gilt*

$$h_n(213, 312) = 2^{\lceil \frac{n}{2} \rceil - 1}.$$

3 Bäume

Beweis. Aus der Tatsache, dass wir das Muster 213 vermeiden, sind die internen Knoten, wie in Satz 3.10, in aufsteigender Reihenfolge markiert. Die Blätter müssen lediglich eine Permutation bilden welche die Muster 213 und 312 vermeiden. Aus Bemerkung 1.6 ergibt sich, dass es dafür $2^{\lceil \frac{n}{2} \rceil - 1}$ Möglichkeiten gibt.

□

Satz 3.15. Für $n \geq 1$ gilt

$$h_n(213, 231, 312) = h_n(213, 231, 321) = h_n(213, 312, 312) = \lceil \frac{n}{2} \rceil.$$

Beweis. Wir werden ein Ergebnis nach dem Anderen berechnen.

a) Nach Satz 3.13 müssen die internen Knoten in aufsteigender Reihenfolge markiert sein. Die größte Markierung n muss auf einem der Blätter auftreten und da wir auch 312 vermeiden, sind auch die Blätter nach n absteigend sortiert. Damit ist bis auf die Position von n , welche $\lceil \frac{n}{2} \rceil$ Möglichkeiten hat, die ganze Permutation fixiert.

b) Wie in a) vermeiden wir 213 und 231. Da wir jedoch zusätzlich 321 vermeiden, müssen die Markierungen der Blätter in aufsteigender Reihenfolge auftreten. Damit ergibt sich das gleiche Ergebnis wie oben.

c) Aus der Tatsache, dass wir 312 und 321 vermeiden, folgt, dass nach n höchstens eine Markierung auftreten darf. Vor n müssen aber wegen 213 alle Markierungen aufsteigend sortiert sein. Damit sind die möglichen Permutationen die Identität id_n , beziehungsweise $\pi_j = 12 \dots (j-1)(j+1) \dots nj$. Da für Heaps die Kindknoten eine größere Markierung tragen müssen als die Eltern Knoten gibt es $\lceil \frac{n}{2} \rceil - 1$ zugelassene Permutationen π_j . Das Ergebnis erhalten wir aus $(\lceil \frac{n}{2} \rceil - 1) + 1 = \lceil \frac{n}{2} \rceil$.

□

Satz 3.16. Sei $n \geq 1$ und F_n bezeichnet die Fibonacci-Zahlen mit $F_1 = 1$ und $F_2 = 1$, dann gilt

$$h_n(231, 312, 321) = F_n.$$

Beweis. Aus den Mustern 312 und 321 folgt, dass n als letzte oder vorletzte Markierung auftritt. Aufgrund von 231 muss die Markierung nach n größer sein als alle davor, woraus folgt, dass nur $(n-1)$ hinter n stehen darf. Nun können wir das Ganze rekursiv darstellen. Denn für einen Heap der Größe $n-1$ erhalten wir einen Heap der Größe n , indem wir ein Blatt mit der Markierung n anfügen. Als zweite Möglichkeit erhalten wir für einen Heap der Größe $n-2$ einen neuen, indem wir zwei neue Blätter anfügen, welche die Markierungen n und $n-1$ tragen. Die beiden Heaps der Größe eins und zwei vermeiden klarerweise die Muster

3 Bäume

und bilden so die Startbedingungen.

□

Je mehr Muster wir vermeiden, desto geringer ist klarerweise die Zahl der Heaps, die übrig bleiben.

Satz 3.17. Für $n \in \{1, 2\}$ gilt $h_n(213, 231, 312, 321) = 1$, für $n \geq 3$ gilt

$$h_n(213, 231, 312, 321) = 2.$$

Beweis. Da wir wieder 312 und 321 vermeiden, muss n als letzte oder vorletzte Markierung auftreten. Wie oben folgt, dass wegen 231 nur $(n-1)$ hinter n stehen darf. Und da wir auch 213 vermeiden, sind die Markierungen vor n in aufsteigender Reihenfolge angeordnet. Es bleiben lediglich die Permutationen id_n und die Permutation $\pi_{n-1} = 12 \dots n(n-1)$ übrig, welche zulässige Permutationen für Heaps sind. Im Fall $n = 1$ beziehungsweise $n = 2$ gibt es nur einen Heap, welcher natürlich alle Muster der Länge drei vermeidet.

□

3.1.3 k-Heaps

Nun wollen wir uns überlegen, welche der Punkte des vorherigen Unterkapitels auf k -Heaps verallgemeinert werden können, so wie in [10] beschrieben. Einige Sätze können direkt übernommen werden, so zum Beispiel Satz 3.8, 3.16 oder 3.17. Einige Andere, so zum Beispiel Satz 3.7, welcher das Muster 123 behandelt muss leicht adaptiert werden. Das Muster wird wie im binären Fall, genau dann vermieden, wenn der Heap die Straucheneigenschaft erfüllt.

Proposition 3.18. Sei $n \geq 1$, dann gilt

$$|\mathcal{H}_n^k(123)| = \begin{cases} 1, & \text{falls } n \leq k + 1 \\ 0, & \text{sonst.} \end{cases}$$

Bemerkung 3.19. Für einige Punkte benötigen wir die Zahl l der Blätter eines k -Heaps mit n Knoten. Diese entspricht

$$l = \left\lceil \frac{(k-1)n - (k-2)}{k} \right\rceil$$

Aus unseren Sätzen 3.9, 3.11, 3.15 erhalten wir entsprechende Aussagen für k -Heaps, wenn wir die Terme $\lceil \frac{n}{2} \rceil$ durch die Zahl der Blätter l ersetzen. Wir wollen Satz 3.9 stellvertretend für alle drei anführen.

3 Bäume

Satz 3.20. Für $n \geq 1$ und die Zahl der Blätter l gilt

$$|\mathcal{H}_n^k(213)| = C_l.$$

Beweis. Wie im Beweis für Satz 3.9 können wir schlussfolgern, dass auf internen Knoten keine Markierungen in absteigender Reihenfolge auftreten dürfen. Desweiteren müssen die Markierungen mit der 1 beginnen und dürfen auch keine Sprünge machen. Damit folgt, dass es für jeden internen Knoten nur eine mögliche Markierung gibt. Die Blätter können beliebig markiert werden, müssen aber das Muster 213 vermeiden. Sei l die Zahl der Blätter so gibt es C_l Möglichkeiten, da es genau diese Anzahl an Permutationen der Länge l gibt, welche dieses Muster vermeiden.

□

3.2 Markierte Bäume

Der nächste Punkt sind die markierten Bäume. In diesem Abschnitt werden wir eine sehr reichhaltige Struktur kennenlernen, welche Mustervermeidung in Permutationen und Mustervermeidung in Strings zusammenführt, woraus eine Vielzahl an Möglichkeiten resultiert. Das Unterkapitel basiert auf [9].

Definition 3.21. Seien X_n Alphabete und $X = \bigcup_{n \geq 2} X_n$, dann bezeichnet ein ebener Wurzelbaum mit Markierungen aus X , oder kurz X -Baum, einen ebenen Wurzelbaum T , für den folgendes gilt. Kein Knoten in T hat genau einen Kindknoten. Jeder interne Knoten v mit genau m Kindern trägt eine Markierung $x_v \in X_m$. Jedes Blatt ist mit einer natürlichen Zahl markiert, wobei folgende zwei Bedingungen gelten müssen.

- i) Jedes der Blätter ist mit einer eindeutigen Zahl von 1 bis l markiert, wobei l die Zahl der Blätter bezeichnet.
- ii) Würden wir jedem internen Knoten die geringste Markierung seiner nachfolgenden Kindknoten zuordnen, dann müssen für jeden internen Knoten v die Markierungen der Kindknoten von links nach rechts ansteigen.

Bemerkung 3.22. Aus Teil ii) der Markierung der Blätter folgt, dass diese Art der Bäume nicht links - rechts - symmetrisch ist.

Beispiel 3.23. Wir betrachten nun einen Baum mit $X = X_2 \cup X_3$, wobei $X_2 = \{o, +\}$ und $X_3 = \{\diamond\}$ gilt.

3 Bäume

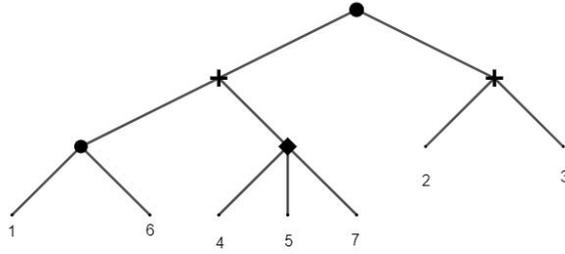


Abbildung 3

Für X wie oben, $i \geq 0$ und $l \geq 1$ bezeichnet $\mathcal{LT}_{i,l}$ die Menge der ebenen Wurzelbäume mit Markierungen aus X , welche i interne Knoten und l Blätter besitzen. Es gilt

$$\mathcal{LT}_l = \bigcup_{i \geq 0} \mathcal{LT}_{i,l}.$$

Die gesamte Menge dieser Bäume wird mit \mathcal{LT} bezeichnet.

Bemerkung 3.24. Der Einfachheit halber werden wir uns auf den Fall $X = X_d$ beschränken. Damit hat jeder interne Knoten d Kindknoten und es wird leichter, explizite Ergebnisse zu berechnen, denn für k interne Knoten muss es $kd - k + 1$ Blätter geben.

Proposition 3.25. Sei $|X_d| = n$, dann gilt

$$|\mathcal{LT}_{kd-k+1}| = n^k \frac{(kd)!}{(d!)^k \cdot k!}.$$

Definition 3.26. Ein Teilbaum S eines Baums $T \in \mathcal{LT}$ entspricht einem internen Knoten und all seinen Nachfolgern. Damit ist abgesichert, dass jeder interne Knoten eine korrekte Markierung beibehalten kann. Punkt ii) der Definition 3.21 bleibt erhalten, jedoch nicht unbedingt Punkt i). Den Baum $\text{st}(S)$ oder auch Standardisierung von S , erhält man durch Änderung der Markierung der Blätter von S . Diese werden ordnungserhaltend durch Elemente aus $\{1, \dots, l\}$ ersetzt, wobei l die Zahl der Blätter bezeichnet. Damit gilt $\text{st}(S) \in \mathcal{LT}$.

Nun, da wir die Basics geklärt haben, beginnen wir mit unserer eigentlichen Arbeit, der Mustervermeidung. Im Zusammenhang der markierten Bäume sprechen wir von einem zusammenhängenden Muster.

Definition 3.27. Ein Baum $T \in \mathcal{LT}$ enthält ein Muster $P \in \mathcal{LT}$ genau dann, wenn es einen Teilbaum S von T gibt, sodass $\text{st}(S) = P$ gilt. Ansonsten sprechen wir davon, dass T das Muster vermeidet.

3 Bäume

Sei \mathcal{P} eine Menge von Mustern aus \mathcal{LT} , dann bezeichnet $\mathcal{LT}(\mathcal{P})$ die Teilmenge der ebenen Wurzelbäume mit Markierungen aus X , welche alle Muster in \mathcal{P} vermeiden. (resp. $\mathcal{LT}_l(\mathcal{P})$ und $\mathcal{LT}_{i,l}(\mathcal{P})$)

Es gilt $f_l(\mathcal{P}) = |\mathcal{LT}_l(\mathcal{P})|$ und die dazugehörige exponentiell erzeugende Funktion ist definiert als

$$F_{\mathcal{P}}(x) = \sum_{l \geq 1} \frac{f_l(\mathcal{P})x^l}{l!}.$$

Desweiteren benötigen wir die exponentiell erzeugende Funktion $f^{\mathcal{P}}(x)$ einer Menge von Mustern aus \mathcal{LT} ,

$$f^{\mathcal{P}}(x) = \sum_{l \geq 1} \frac{|\mathcal{P} \cap \mathcal{LT}_l| x^l}{l!}.$$

Dazu kommt die erzeugende Funktion des Markenvorrats X

$$f_X(x) = \sum_{n \geq 1} \frac{|X_n| x^n}{n!}.$$

Wir verwenden wie üblich bei markierten Objekten, exponentiell erzeugende Funktionen. Die folgende Proposition zeigt, dass die Kettenregel wie gewohnt der Substitution entspricht.

Proposition 3.28. *Seien K und L zwei Mengen aus \mathcal{LT} über dem Alphabet X . Sei nun die Menge M definiert, als die Menge jener Bäume, welche auf folgende Weise konstruiert sind. Ein Element aus K bildet die Grundlage. Doch anstatt der Blätter werden Elemente der Menge L angefügt. Die exponentiell erzeugende Funktionen verhalten sich wie folgt*

$$f^M(x) = f^K(f^L(x)). \quad (23)$$

3.2.1 Die Cluster Inversions Formel

Definition 3.29. Sei P eine Menge von Mustern aus \mathcal{LT} . Ein Baum T mit einer Menge an Teilbäumen T_1, \dots, T_k wird k -Cluster genannt, falls folgende zwei Bedingungen gelten.

1. Für alle i gilt: $\text{st}(T_i) \in P$.
2. Jede Kante, welche zwei interne Knoten von T verbindet, ist eine Kante aus T_i , welche zwei interne Knoten von T_i miteinander verbindet.

Die Menge aller k -Cluster über einer Menge P wird mit $c_{n,k}(P)$ bezeichnet, wobei n die Zahl der Blätter angibt.

3 Bäume

Über die k -Cluster können wir nun die Inverse $F_{\mathcal{P}}^{-1}(x)$ zu einer erzeugenden Funktion $F_{\mathcal{P}}(x)$ berechnen, sodass $F_{\mathcal{P}}^{-1}(F_{\mathcal{P}}(x)) = x$ gilt.

Satz 3.30. *Für eine Menge von Mustern \mathcal{P} gilt*

$$F_{\mathcal{P}}^{-1}(x) = x - \sum_{n \geq 0, k \geq 0} \frac{(-1)^k c_{n,k}(\mathcal{P}) x^n}{n!}. \quad (24)$$

Dieser Satz kann direkt über das Inklusions-Exklusionsprinzip bewiesen werden, aber auch über Homologische Algebren siehe [9].

Korollar 3.31. *Wir betrachten nun den Fall $X = X_2$ für Muster mit drei Blättern. Sei \mathcal{P} eine beliebige Menge von Mustern aus \mathcal{LT}_3 und \mathcal{P}^c dessen Komplement, so gilt*

$$F_{\mathcal{P}}(-F_{\mathcal{P}^c}(-x)) = F_{\mathcal{P}^c}(-F_{\mathcal{P}}(-x)) = x. \quad (25)$$

Der Beweis hierfür ist in [9] zu finden. Bevor wir zu Anwendungen kommen, benötigen wir noch eine bestimmte Art von Bäumen, welche auch in späteren Kapiteln immer wieder vorkommen wird.

Definition 3.32. Ein Baum t wird left-comb (resp. right-comb) genannt, falls für jeden Knoten aus t folgendes gilt. Der linkeste (resp. rechteste) Kindknoten kann ein interner Knoten oder ein Blatt sein, alle anderen Kindknoten sind zwangsweise Blätter.

Man beachte, dass man am Beispiel von left- beziehungsweise right-comb die Asymmetrie der ebenen Wurzelbäume mit Markierungen aus X besonders schön erkennen kann.

Beispiel 3.33. Für $X = X_2$ und $|X| = 1$ betrachten wir nun die Bäume left-comb mit n internen Knoten und $n + 1$ Blätter. Für die Markierung der Blätter gilt folgendes. Das linkeste Blatt muss die Markierung 1 tragen. Die Markierung aller anderen Knoten ist beliebig, da durch Bedingung *ii*) der Definition 3.21 jeder interne Knoten die Markierung 1 bekommt. Das bedeutet die Blattmarkierungen Bäume left-comb $_n$ können als Permutationen der Länge n interpretiert werden.

Beispiel 3.34. Nun betrachten wir die Bäume right-comb mit n internen Knoten. Hier gilt, genau das umgekehrte Extrem. Es gibt nur eine einzige Möglichkeit Punkt *ii*) der Definition 3.21 zu erfüllen, aufsteigend. Die Markierungen der internen Knoten sind jedoch beliebig. Damit kann jeder Baum right-comb als Wort aus dem Alphabet X_2 interpretiert werden.

3 Bäume

Aufgrund dieses Beispiels ist einerseits die Asymmetrie der Bäume klar, andererseits kann man auch erkennen, warum diese kombinatorische Struktur so reichhaltig ist. Sie beinhaltet, wie wir gerade gesehen haben, die Mustervermeidung in Permutationen, genau so wie die Mustervermeidung im Bereich der Wörter über einem Alphabet. Zu guter letzt lässt der Markenvorrat X großen Spielraum.

3.2.2 Asymptotisches Verhalten

In diesem Abschnitt beschäftigen wir uns mit dem Versuch, Aussagen über das asymptotische Verhalten der Bäume zu finden. Wir beginnen mit der Adaptierung der Golod-Shafarevich-Ungleichung.

Satz 3.35. *Sei P eine Menge von Mustern, so gilt die folgende Ungleichung koeffizientenweise*

$$f^P(F_P(x)) - f_X(F_P(x)) + F_P(x) \geq x.$$

Beweis. Wir betrachten dazu die endlichen Mengen B_n , der Bäume aus \mathcal{LT}_n , welche den Bäumen des ersten Terms entsprechen. Also Bäume mit beliebiger Wurzel und nachfolgenden Teilbäumen, welche Muster in P vermeiden. Zusätzlich betrachten wir die endlichen Mengen $C_n \subseteq P \times \mathcal{LT}_n$, der Form (P, T) . Wobei gelten muss, dass es einen Teilbaum S von T gibt, wessen Wurzel mit der von T zusammenfällt und $\text{st}(S) = P$ gilt. Alle Teilbäume, welche an S anhängen, vermeiden Muster aus P . Damit ist unsere zu zeigende Gleichung für $n \geq 2$ koeffizientenweise äquivalent zu

$$|C_n| - |B_n| + |\mathcal{LT}(P)| \geq 0.$$

Diese Aussage ist wahr, da es nach konstruktion klarerweise eine surjektive Abbildung von $C_n \cup \mathcal{LT}_n(P)$ nach B_n gibt. Denn Bäume in B_n vermeiden entweder die Muster P oder enthalten sie an der Wurzel.

□

Satz 3.36. *Sei $|X| = 1$ und $X = X_d$ für ein beliebiges $d \geq 2$. Das Muster P hat $K \geq 2$ interne Knoten, dann gibt es für alle Paare (d, k) außer $(2, 2), (2, 3), (2, 4)$ und $(3, 2)$ eine positive Zahl C , sodass folgendes gilt:*

$$|\mathcal{LT}_n(P)| \geq C^{n-1} |\mathcal{LT}_n|.$$

Der Beweis dieses Satzes ist das Hauptresultat von Paper [9] und würde den Rahmen dieser Arbeit sprengen. Die Autoren veranlasste er, folgende Definition aufzustellen.

3 Bäume

Definition 3.37. Die Wachstumsrate einer Menge von Mustern \mathcal{P} ist definiert als

$$\limsup_{n \rightarrow \infty} \left(\frac{|\mathcal{LT}_n(\mathcal{P})|}{|\mathcal{LT}_n|} \right)^{\frac{1}{n-1}}. \quad (26)$$

Dem entsprechend und in der Annahme, dass der obige Satz leicht auf einen beliebigen Markenvorrat X erweiter werden könne, stellen sie folgende Vermutung auf.

Vermutung 3.38. Sei X ein beliebiger Markenvorrat, dann existiert eine Zahl d , sodass für jedes Muster P in \mathcal{LT} mit mindestens d internen Knoten und Zahlen $c(P) > 0, \lambda(P) > 1$

$$\frac{|\mathcal{LT}_n(P)|}{|\mathcal{LT}_n|} \sim c(P)\lambda(P)^{-n}$$

gilt.

3.3 Binärbäume

Hier im zweiten Teil dieses Kapitels bearbeiten wir nichtmarkierte Bäume. Wir starten mit den allseitsbekannten Binärbäumen und werden dann versuchen, das Erarbeitete auch auf allgemeinere Bäume zu übertragen, die k - Bäume. Zu Beginn dreht sich alles um zusammenhängende Teilbäume. Später jedoch werden wir auch noch unzusammenhängende Teilbäume näher betrachten. Wir starten mit einem Abschnitt analog zu [11].

Definition 3.39. Ein Binärbaum ist ein ebener Wurzelbaum, wobei jeder Knoten entweder null oder zwei Kindknoten hat. Die Menge aller Binärbäume wird mit \mathcal{T} bezeichnet und \mathcal{T}_n bezeichnet die Binärbäume mit n internen Knoten.

Bemerkung 3.40. Ein beliebiger Binärbaum mit n internen Knoten hat $n + 1$ Blätter.

Definition 3.41. Seien s und t zwei binäre Baume. Wir sprechen davon, dass der Baum s den Baum t als Muster enthält, falls t ein ebener zusammenhängender Teilbaum von s ist. Andernfalls sprechen wir davon, dass s das Muster t vermeidet.

Nun können wir auf der Menge der Binärbaume eine binäre Operation definieren.

$$\begin{aligned} \wedge : \mathcal{T} \times \mathcal{T} &\rightarrow \mathcal{T} \\ (s, t) &\rightarrow s \wedge t \end{aligned}$$

3 Bäume

Der Ausdruck $s \wedge t$ entspricht folgendem Binärbaum. Es gibt die eindeutige Wurzel mit zwei nachfolgenden Teilbäumen. Der linke Teilbaum entspricht s und der Rechte t .

Sei nun $*$ eine beliebige binäre Operation und x_0, \dots, x_n Variablen, dann ergibt der Ausdruck

$$x_0 * x_1 * \dots * x_n$$

erst mit einer Klammersetzung Sinn. Hier kommen die Binärbäume ins Spiel, denn jede Klammersetzung kann als eigener Binärbaum interpretiert werden. Ersetzt man in einem beliebigen geklammerten Ausdruck, zum Beispiel $((x_0 * (x_1 * x_2)) * x_3)$ jedes Vorkommen des Symbols $*$ durch ein \wedge , jede der Variablen x_i durch einen Baum t_i , welcher nur aus einem Knoten besteht, so erhält man einen eindeutigen Binärbaum

$$((t_0 \wedge (t_1 \wedge t_2)) \wedge t_3).$$

Natürlich kann auch jeder Binärbaum aus \mathcal{T}_n als Klammersatz für $n + 1$ Variablen interpretiert werden. Nach Bemerkung 2.3 kann damit auch jeder Binärbaum als n -Dyck-Pfad interpretiert werden. Eine Bijektion zwischen diesen beiden Strukturen werden wir später noch kennen lernen. Fürs Erste begnügen wir uns mit der Folgerung, dass die Zahl der Binärbäume mit n internen Knoten gleich der Catalan-Zahl C_n sein muss.

Weiter geht es mit der üblichen Notation, welche wir für die Mustervermeidung benötigen. Sei A_n die Zahl der Binärbäume mit n internen Knoten, welche ein gegebenes Muster t vermeiden. Die Zahl $A_{n,k}$ bezeichnet die Zahl jener Bäume, welche genau k Kopien des Musters t enthalten. Damit lassen sich die dazugehörigen erzeugenden Funktionen wie folgt darstellen

$$\begin{aligned} \text{Av}_t(x) &= \sum_{n \geq 0} A_n x^n, \\ \text{En}_t(x, y) &= \sum_{n \geq 0} \sum_{k \geq 0} A_{n,k} x^n y^k. \end{aligned}$$

Klarerweise gilt $\text{Av}_t(x) = \text{En}_t(x, 0)$. Die Autoren von [11] stellten folgende Vermutung auf [11, S. 745, Vermutung 2]

Vermutung 3.42. *Für zwei Binärbäume mit n internen Knoten s und t folgt aus $\text{Av}_t(x) = \text{Av}_s(x)$, dass $\text{En}_t(x, y) = \text{En}_s(x, y)$ gilt.*

Oder anders formuliert, zwei gleich große Muster, welche in gleich vielen Bäumen aus \mathcal{T}_n vermieden werden, kommen in gleich vielen Bäumen in \mathcal{T}_n k mal vor.

3 Bäume

Nun wollen wir mit einigen einfachen Aussagen beginnen. Sei L der Binärbaum welcher nur aus einem einzigen Blatt besteht. Klarerweise muss jeder Baum mindestens einen Knoten enthalten, darum gilt $Av_L(x) = 0$. Jeder Binärbaum mit n internen Knoten enthält $n + 1$ Blätter, damit muss

$$En_L(x, y) = \sum_{n \geq 0} C_n x^{2n+1} y^{2n+1}$$

gelten. Die etwas interessanteren Beispiele treten ab drei internen Knoten auf. Hier gibt es fünf verschiedene Bäume.

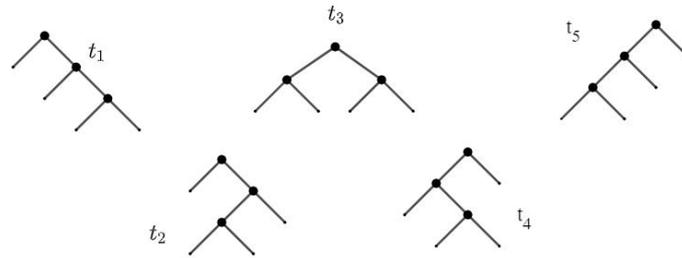


Abbildung 4

Die Bäume t_1 (t_5) sind eine Form der left- (right-) combs. Sie werden noch häufiger vorkommen, darum werden wir hier die etwas genauere Notation einführen. Der Baum left-comb_0 ist definiert als der Baum L , welcher nur aus einem Knoten besteht.

$$\text{left-comb}_{k+1} := \text{left-comb}_k \wedge L \quad (27)$$

Die Bäume right-comb_k werden auf die gleiche Weise definiert wie left-comb_k . Die Bäume t_1 resp. t_5 entsprechen den Bäumen left-comb_3 resp. right-comb_3 . Der Baum left-comb_k^1 ist definiert als $L \wedge \text{left-comb}_k$. Der Baum t_2 entspricht left-comb_2^1 .

Ab dieser Komplexitätsstufe treten nämlich zum ersten Mal (nicht-triviale) Wilf-Äquivalenzklassen auf. So bilden die Bäume t_1 und t_5 eine Äquivalenzklasse. Die zweite bilden die Bäume t_2, t_3 und t_4 .

Um dies zu zeigen, benötigen wir eine Bijektion γ zwischen den Binärbäumen mit n internen Knoten und den gewöhnlichen ebenen Bäumen mit $n + 1$ Knoten.

3 Bäume

Definition 3.43. Die Abbildung $\gamma : \mathcal{T}_n \rightarrow \mathcal{PT}_{n+1}$ ist definiert als jene Funktion, welche auf einem Binärbaum jede Kante zu linken Kindern kontrahiert.³

Das Bild eines Binärbaums aus \mathcal{T}_n ist ein ebener Baum und es gilt, dass jeder Knoten zwischen null und n Kindknoten haben kann.

Lemma 3.44. Die Abbildung γ ist eine Bijektion.

Dieser Beweis wird später nachgebracht (siehe Lemma 4.20).

Satz 3.45. Die erzeugende Funktion für Bäume, welche das Muster t_1 (resp. t_5) vermeiden entspricht

$$\text{Av}_{t_1}(x) = \text{Av}_{t_5}(x) = \sum_{n \geq 0} M_n x^{2n+1}.$$

Beweis. Der erste Teil der Aussage ist klar, da die Bäume t_1 und t_5 die links - rechts Spiegelung von einander sind. Den zweiten Teil zeigen wir, indem wir eine Bijektion von den Binärbäumen mit n internen Knoten, welche das Muster t_5 vermeiden, zur Menge der Motzkin-Pfade \mathcal{M}_n konstruieren.

Zunächst wenden wir die oben beschriebene Funktion γ an. Doch um zu verstehen wie die Bilder von Binärbäumen, welche t_5 vermeiden, unter γ aussehen ist wichtig, dass wir wissen wie das Bild von t_5 unter γ aussieht. Der Baum $\gamma(t_5)$ hat klarerweise vier Knoten und durch einfache Überlegungen, oder wie wir auch später besser begründen können, stellen wir fest, dass die Wurzel drei Kindknoten hat. Daraus folgern wir, dass ein Binärbaum genau dann t_5 vermeidet, falls sein Bild unter γ keinen Knoten mit drei oder mehr Kindknoten hat.

Nun können wir für jeden solchen Baum eine Abfolge aus dem Alphabet $\{-1, 0, 1\}$ konstruieren. Für jeden Knoten, den wir nach und nach bei der Tiefensuche antreffen würden, hängen wir die Zahl $m - 1$ an das bereits bestehende Wort an, wobei m die Zahl der Kindknoten dieses Knotens ist. Schlussendlich löschen wir noch die letzte Zahl, welche eine -1 sein muss.

Man überlegt sich leicht, dass das so entstandene Wort wirklich einer Exkursion entsprechen muss. Die Funktion ist injektiv, denn zwei verschiedene ebene Bäume müssen sich an mindestens einer Stelle in der Zahl der Kindknoten unterscheiden. Klarerweise haben deswegen beide an dieser Stelle eine andere Zahl in ihrem zugehörigen Motzkin Wort. Auch die Surjektivität kann leicht gezeigt werden, da wir zu einem bestehenden Wort sofort den dazugehörigen Baum konstruieren können.

□

³Kontraktion sieht Appendix 6.3

3 Bäume

Wir widmen uns der zweiten Wilf-Äquivalenzklasse und starten beim Baum t_3 . Ein Baum vermeidet dieses Muster, falls kein Knoten zwei Kindknoten besitzt, welche jeweils wieder zwei Kindknoten besitzen. Damit muss jeder interne Knoten mindestens ein Blatt als Kindknoten haben. Ein Baum mit n internen Knoten hat $n + 1$ Blätter. Zwei davon müssen auf der untersten Ebene liegen. Die restlichen $n - 1$ haben die zwei Möglichkeiten das linke oder das rechte Kind zu sein. Wir können einen solchen Baum als ein Wort der Länge $n - 1$ aus dem Alphabet $\{L, R\}$ interpretieren, damit gibt es 2^{n-1} verschiedene Bäume, welche t_3 vermeiden. Die erzeugende Funktion sieht folgendermaßen aus

$$\text{Av}_{t_3}(x) = x + x^3 + 2x^5 + \dots + 2^{n-1}x^{2n+1} = x + \sum_{n \geq 0} 2^n x^{2n+3} = \frac{x(1-x^2)}{1-2x^2}.$$

Satz 3.46. *Die Bäume t_2, t_3 und t_4 liegen in der gleichen Wilf-Äquivalenzklasse.*

Beweis. Aufgrund der links - rechts - Symmetrie ist die Relation von t_2 und t_4 klar. Zu zeigen ist, dass es eine Bijektion zwischen den Bäumen, welche t_3 vermeiden, und den Bäumen, welche t_2 vermeiden, gibt. Denn auch Binärbäume, welche das Muster t_2 vermeiden können durch ein Wort der Länge $n - 1$ aus dem Alphabet $\{L, R\}$ interpretiert werden.

Für einen Baum, welcher t_2 vermeidet, muss gelten, dass jedes rechte Kind jedes Knotens nur das Blatt L als linkes Kind haben darf. Damit entsteht folgende Struktur. Von der Wurzel gibt es einen Teilbaum, oder auch Spine genannt, welcher einem left-comb_k entspricht. Alle anderen Knoten sind die rechten Kinder eines Knotens und dürfen daher nur mehr Blätter als linken Nachfolger haben. Damit gibt es k verschiedene Teilbäume, alles die rechten Nachfolger eines Knotens in der Spine, welche alle right-comb_{n_i} entsprechen. Die Darstellung aus dem Alphabet $\{L, R\}$ sieht nun folgendermaßen aus

$$R^{n_0} L R^{n_1} L \dots L R^{n_{k-1}}.$$

Für einen Baum mit n internen Knoten hat dieses Wort die Länge $n - 1$, da jeder interne Knoten außer der Wurzel einen Buchstaben beiträgt. Ebenso kann jedes Wort, nach der obigen Beschreibung, als solcher Baum interpretiert werden.

□

Um fortzufahren, benötigen wir wieder eine Bijektion, diesmal zu den Dyck-Pfaden. Die Abbildung ϵ bildet ebene Bäume mit $n + 1$ Knoten auf n -Dyck-Pfade folgendermaßen ab. Wir erzeugen ein Dyck-Wort, indem wir den Baum durchlaufen und immer wenn wir eine Kante von einem Knoten zu seinem Kind passieren,

3 Bäume

hängen wir eine U an das bereits bestehende Wort an. Immer wenn wir eine Kante von einem Kindknoten zum Elternknoten durchlaufen, so hängen wir eine D an das Wort an. Der Baum wird beginnend mit der Wurzel durchlaufen und nach jedem Knoten gehen wir zum linkensten noch nicht abgearbeiteten Knoten und falls wir alle Kindknoten abgearbeitet haben oder wir bei einem Blatt angelangt sind, so gehen wir zum Elternknoten. Nachdem wir jede Kante genau zwei mal passiert haben, wurde das Dyck-Wort erzeugt und der Algorithmus terminiert.

Lemma 3.47. *Die oben beschriebene Funktion $\epsilon : \mathcal{PT}_{n+1} \rightarrow \mathcal{D}_n$ ist wohldefiniert und eine Bijektion.*

Beweis. Klarerweise wird durch diesen Algorithmus für jeden Baum ein Wort der Länge $2n$ aus dem Alphabet $\{U, D\}$ erzeugt. Da wir für jede Kante genau einmal ein U und einmal ein D erhalten, ist der erste Teil von Bemerkung 2.2 erfüllt. Für den zweiten Teil überlegen wir uns, dass wir bei der Wurzel starten. Damit erhalten wir von einer Kante erst ein D, falls wir bereits ein U erhalten, woraus folgt, dass vor jedem i-ten D bereits i Us stehen müssen.

Die Injektivität erhalten wir durch die Überlegung, dass zwei verschiedene Bäume sich mindestens ein Knotengrad unterscheiden muss. Kommen wir durch unseren Algorithmus zu diesem Punkt, so unterscheidet sich dort das erzeugte Wort. Die Surjektivität folgt aus der Injektivität und der Tatsache, dass beide Strukturen durch die Catalan-Zahlen C_n gezählt werden.

□

Interessant an dieser Abbildung ist, dass jedes Blatt im Baum einem Peak im Dyck-Pfad entspricht beziehungsweise jedes Blatt auf der k-ten Ebene unter der Wurzel entspricht einem Peak auf Höhe k. Damit könnten wir einige interessante Zusammenhänge aus dem Kapitel über Peaks auf Bäume übertragen.

Mit der Abbildung ϵ erhalten wir auch eine Bijektion zwischen den Binärbäumen und den Dyck-Pfaden.

Definition 3.48. Die Abbildung $\delta : \mathcal{T}_n \rightarrow \mathcal{D}_n$ ist definiert als

$$\delta := \epsilon \circ \gamma. \tag{28}$$

Als Verkettung von zwei Bijektionen ist δ natürlich wieder eine Bijektion. Nun können wir versuchen, weitere Zusammenhänge aus der Mustervermeidung in Dyck-Pfaden und der Mustervermeidung in Binärbäumen zu finden.

Beispiel 3.49. Der Baum t_1 entspricht unter γ einer Kette aus vier Knoten. Das Bild unter δ , also das entsprechende Dyck-Wort, sieht folgendermaßen aus: $UUUDDD$. Nun kann man sich überlegen, ob es einen Zusammenhang zwischen

3 Bäume

Bäumen, welche t_1 vermeiden und Dyck-Pfaden, welche ein bestimmtes Muster vermeiden, gibt.

In der Tat lässt sich zeigen, dass n -Dyck-Pfade, welche das Muster UUU vermeiden, in Bijektion mit Bäumen, welche t_1 vermeiden, stehen. Einen Beweis und eine noch stärkere Aussage gibt es im nächsten Kapitel mit Satz 4.33.

Wir wollen das obige Beispiel verallgemeinern und fragen uns ob und wann es Bijektionen von Bäumen welche ein Muster t vermeiden und Dyck-Pfaden welche ein gewisses Muster vermeiden gibt. Antwort darauf liefert uns folgendes Zitat [11, S. 748]

'In general there is a bijection between n -leaf binary trees avoiding t and $(n-1)$ -Dyck words avoiding w whenever w is a characteristic feature of $\gamma(t)$, that is, some feature of the tree that is preserved locally by γ '.

Weitere Beispiele für solche Verbindungen sind die Bäume t_4 , left-comb_3^1 oder $L \wedge (\text{right-comb}_3 \wedge L)$. Diese stehen mit Dyck-Pfaden in Bijektion welche die Muster UDD , $UDDD$ beziehungsweise $UUDDD$ vermeiden. Mit dem Wissen aus dem Kapitel über Gitterpfade können wir die erzeugende Funktion für alle diese Beispiele (bis auf $UDDD$) leicht berechnen.

Wir setzen mit dem Versuch fort, für jedes Muster t die erzeugende Funktion $Av_t(x)$ zu finden. Wir sind zwar leider nicht in der Lage, diese explizit anzugeben, jedoch werden wir einen Algorithmus kennen lernen, um eine Funktionalgleichung für $Av_t(x)$ aufzustellen. Dieser Algorithmus funktioniert auch für allgemeinere k -Bäume, wobei sich abgesehen von den offensichtlichen Dingen nichts ändert. Der Einfachheit halber werden wir ihn jedoch nur für Binärbäume vorstellen. Hierfür benötigen wir die Menge aller Binärbäume t , welche ein Muster p als Teilbaum beinhalten, wobei zusätzlich gefordert ist, dass die Wurzel von p mit der Wurzel von t übereinstimmt. Diese Menge wird mit $\mathcal{L}(p)$ bezeichnet. Klarerweise gilt $\mathcal{L}(L) = \mathcal{T}$. Desweiteren benötigen wir den Durchschnitt $\mathcal{L}(p) \cap \mathcal{L}(q)$. Das sind jene Bäume, welche die beiden Muster p und q an der Wurzel beinhalten. Falls p (resp. q) dem Baum, welcher nur aus einem Knoten besteht, L entspricht, so gilt $\mathcal{L}(L) \cap \mathcal{L}(q) = \mathcal{L}(q)$. Für $p = p_L \wedge p_R$ und $q = q_L \wedge q_R$ gilt $\mathcal{L}(p) \cap \mathcal{L}(q) = (\mathcal{L}(p_L) \cap \mathcal{L}(q_L)) \wedge (\mathcal{L}(p_R) \cap \mathcal{L}(q_R))$, wobei $A \wedge B := \{a \wedge b : a \in A, b \in B\}$.

Definition 3.50. Sei t ein festes Muster. Für einen Binärbaum p ist das Gewicht von $\mathcal{L}(p)$ definiert als

$$\text{weight}(\mathcal{L}(p)) := \sum_{T \in \mathcal{L}(p)} \text{weight}(T). \quad (29)$$

3 Bäume

Wobei das Gewicht eines Baumes definiert ist als

$$\text{weight}(T) := \begin{cases} x^{\text{Zahl der Knoten in } T}, & \text{falls } T \text{ das Muster } t \text{ vermeidet} \\ 0, & \text{sonst.} \end{cases}$$

Nun benötigen wir nur mehr eine simple Überlegung, bevor wir zum eigentlichen Algorithmus kommen. Der Fall $t = L$ wurde bereits abgehandelt. Wir kommen nun zum Fall $t = t_L \wedge t_R$. Für das Gewicht von $p = p_L \wedge p_R$ gilt

$$\begin{aligned} \text{weight}(\mathcal{L}(p)) = \text{weight}(\mathcal{L}(p_L \wedge p_R)) = & x(\text{weight}(\mathcal{L}(p_L))\text{weight}(\mathcal{L}(p_R)) - \\ & - \text{weight}(\mathcal{L}(p_L) \cap \mathcal{L}(t_L))\text{weight}(\mathcal{L}(p_R) \cap \mathcal{L}(t_R))). \end{aligned} \quad (30)$$

Die Idee ist klar, wir versuchen ein Muster rekursiv zu zerteilen um auf mehrere kleinere Teilaufgaben zu kommen. Die Gleichung (30) entstammt der Überlegung, dass eine Muster genau dann vermieden wird, falls der linke beziehungsweise der rechte Teilbaum das Muster vermeidet oder das in t_L und t_R zerteilte Muster vom zugehörigen (linken / rechten) Teilbaum vermieden wird. Wir multiplizieren das Ganze mit x , da beim Zerteilen des Baums ein Knoten, die Wurzel, verloren geht und sonst das Gewicht reduziert werden würde.

Für den Algorithmus starten wir mit der Gleichung

$$\text{weight}(\mathcal{L}(L)) = \text{weight}(L) + \text{weight}(\mathcal{L}(L \wedge L)). \quad (31)$$

Denn es gilt $\text{weight}(\mathcal{L}(L)) = \text{Av}_t(x)$ und $\text{weight}(L) = x$. Alle anderen Terme werden mit Gleichung (30) auf einfachere zerteilt. Schlussendlich erhalten wir nach mehrfacher Anwendung von Gleichung (30) eine einzelne Polynomgleichung in den Variablen x und $\text{Av}_t(x)$.

Beweis. Wir starten mit der Gleichung (31). Im nächsten Schritt wird der Term $\text{weight}(\mathcal{L}(L \wedge L))$ mithilfe von Gleichung (30) zerteilt. Dadurch können potenziell zwei neue Terme entstehen: t_L und t_R . Für jeden dieser Terme und für jeden weiteren der entsteht wenden wir Gleichung (30) an. Wichtig ist, dass wir für zwei Bäume p, q den Durchschnitt $\mathcal{L}(p) \cap \mathcal{L}(q)$ berechnen können und es einen Baum s gibt, sodass dieser gleich $\mathcal{L}(s)$ ist. Für den Baum s gilt, dass er höchstens so viele Ebenen wie das Maximum von p resp. q hat. Daraus folgt, dass nur endlich viele Terme auftreten können und der Algorithmus terminiert. Nun müssen wir die erhaltenen Gleichungen auflösen, zum Beispiel indem wir eine Gröbner Basis berechnen. Dadurch erhalten wir eine einzelne Gleichung in x und $\text{Av}_t(x)$.

□

3 Bäume

Ganz analog verläuft auch der Versuch, eine Gleichung für $\text{En}_t(x, y)$ zu finden. Das einzige, das verändert werden muss, ist das Gewicht eines Baumes T

$$\text{weight}(T) = x^{\text{Zahl der Knoten in } T} y^{\text{Zahl der Kopien von } t \text{ in } T} \quad (32)$$

und Gleichung (30). Diese wird zu

$$\begin{aligned} \text{weight}(\mathcal{L}(p)) = \text{weight}(\mathcal{L}(p_L \wedge p_R)) = & x(\text{weight}(\mathcal{L}(p_L))\text{weight}(\mathcal{L}(p_R)) + \\ & +(y - 1)\text{weight}(\mathcal{L}(p_L) \cap \mathcal{L}(t_L))\text{weight}(\mathcal{L}(p_R) \cap \mathcal{L}(t_R))). \end{aligned} \quad (33)$$

Die Veränderung der Gleichung erscheint wohl logisch, da wir, falls das Muster aufgeteilt auf die Bäume p_L und p_R vorkommt, durch y gezählt werden. Klarerweise gilt auch, dass wir für $y = 0$ den etwas einfacheren Algorithmus für $\text{Av}_t(x)$ erhalten. Abgesehen davon bleibt alles beim Alten. Der Algorithmus terminiert nach zahlreicher Anwendung von (33) und wir bekommen eine Gleichung in x, y und $\text{En}_t(x, y)$.

Solche Gleichungen sind nicht immer leicht zu lösen. Jedoch benötigen wir nicht immer die expliziten Funktionen $\text{Av}_t(x)$ beziehungsweise $\text{En}_t(x, y)$. Denn um Wilf-Äquivalenzen zu erkennen, reichen die impliziten Gleichungen für $\text{Av}_t(x)$ aus. Die Autoren von Paper [11] berechnen mithilfe des Algorithmus die Wilf-Äquivalenzklassen der Bäume mit bis zu sechs internen Knoten. Desweiteren berechnet sie die dazugehörigen Gleichungen für $\text{En}_t(x, y)$ und stellen fest, dass für Wilf-äquivalente Bäume s, t auch $\text{En}_s(x, y) = \text{En}_t(x, y)$ gilt. Aus diesem Grund stellen sie Vermutung 3.42 auf.

3.4 k - Bäume

Weiter geht es mit einem Abschnitt über k -Bäume, welcher Inhalte aus [13] verwendet. Hier werden wir versuchen, das bereits im Kapitel über Binärbäume erarbeitete auf k -Bäume zu generalisieren.

Definition 3.51. Ein k -Baum ist definiert als ebener Wurzelbaum, wobei jeder Knoten entweder k Kindknoten besitzt oder ein Blatt ist.

Bemerkung 3.52. Es gilt weiterhin Definition 3.41, wir verwenden jedoch k -Bäume als Muster. Außerdem verwenden wir weiterhin die gleiche Notation wie im Kapitel über Binärbäume.

Wir starten wie gewöhnlich mit einem simplen Fall und werden uns dann auf die etwas komplizierteren Themen stürzen.

3 Bäume

Beispiel 3.53. Für dieses Beispiel betrachten wir 3-Bäume oder auch ternäre Bäume genannt. Insbesondere wollen wir jenen Baum t untersuchen, welcher drei interne Knoten hat. Die Wurzel von t besitzt genau ein Blatt an der Position ihres dritten Kindknotens.

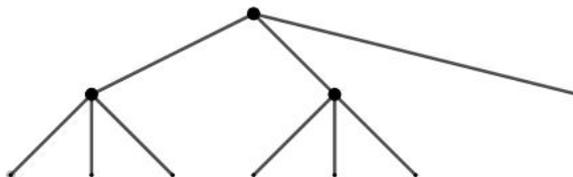


Abbildung 5

Sei nun T ein Baum mit n internen Knoten, welcher das Muster t vermeidet. Für einen Knoten v aus T gilt nun folgendes. Die Kindknoten von v sind ein Teilbaum, welcher t vermeidet und k interne Knoten besitzt, ein Blatt und wieder ein Teilbaum oder der Form $(\text{Blatt}, \text{Baum}, \text{Baum})$. Der Fall $(\text{Blatt}, \text{Blatt}, \text{Baum})$ wird doppelt gezählt muss daher abgezogen werden und es gilt

$$A_n = 2 \sum_{k=1}^{n-2} A_k A_{n-k-1} - A_{n-1}.$$

Durch umformen und mit Hilfe von exponentiell erzeugenden Funktionen kommen wir auf die Gleichung

$$(1+x)Av_t(x) - 2(Av_t(x))^2 - x = 0.$$

Woraus sich die explizite Lösung

$$Av_t(x) = \frac{1+x - \sqrt{x^2 - 6x + 1}}{4x}$$

ergibt. Dies entspricht der erzeugenden Funktion der kleinen Schröder-Zahlen beziehungsweise der Hälfte der großen Schröder-Zahlen aus Beispiel 6.3.

Für den kommenden Abschnitt über die Wortdarstellung eines Baumes benötigen wir noch eine kleine Definition.

Definition 3.54. Für einen k -Baum t bezeichnet der Ausdruck k -Blatt Elternknoten einen Knoten aus t , welcher nur Blätter als Kindknoten besitzt.

3 Bäume

Ziel der Wortdarstellung ist es jeden k -Baum eindeutig als eine Menge aus Strings darstellen zu können.

Definition 3.55. Die Wort Darstellung M eines k -Baums t ist durch durch folgenden Algorithmus beschrieben. Zunächst werden die Kindknoten jedes Knotens von links nach rechts mit 1 bis k markiert. Danach wird für jeden Pfad von der Wurzel zu einem k Blatt Elternknoten ein String in die Menge M hinzugefügt, wobei der String der Abfolge der Markierungen der Knoten des Pfades entspricht.

Nun kann man sich überlegen wie die Wortdarstellungen von zwei Bäumen s und t aussehen, wenn gilt, dass s ein Teilbaum von t ist. Sie $\{s_i\}_{i \in I}$ (resp. $\{t_i\}_{i \in J}$) die Wortdarstellung von s (resp. t). Dann muss es einen möglicherweise leeren Präfix p_s geben, sodass für jedes $i \in I$ der String $(p_s s_i)$ Präfix eines Strings t_j ist.

Diese Wortdarstellung vereinfacht nicht nur das Überprüfen ob ein Baum, Teilbaum des anderen ist und hilft beim Pattern-Matching, sondern kann auch dafür verwendet werden verschiedene Wilf-Äquivalenzen zu finden.

Proposition 3.56. Sei k fest und $b : \{1, \dots, k\} \rightarrow \{1, \dots, k\}$ eine Bijektion. Weiters sei $[k]^n$ die Menge aller Strings der Länge n über dem Alphabet $\{1, \dots, k\}$, dann gibt es eine eindeutig bestimmte Fortsetzung $B : [k]^n \rightarrow [k]^n$ von b . Zusätzlich gibt es eine eindeutig bestimmte Fortsetzung \hat{B} von B , welche auf Mengen von Strings definiert ist.

Lemma 3.57. Seien s und t zwei k -Bäume mit der einelementigen Darstellung $\{s_0\}$ und $\{t_0\}$. Angenommen es gibt eine Bijektion $b_{s,t} : \{1, \dots, k\} \rightarrow \{1, \dots, k\}$ mit Fortsetzung $B_{s,t}$. Wir nehmen zusätzlich an, dass B den String s_0 auf t_0 abbildet, dann gibt es eine Bijektion $\hat{B}_{s,t}$ zwischen den k -Bäumen, welche das Muster s vermeiden und den k -Bäumen welche das Muster t vermeiden.

Beweis. Die obige Proposition sichert die Existenz der Abbildung $\hat{B}_{s,t}$. Klarerweise gibt es auch die Inverse Abbildung $\hat{B}_{s,t}^{-1}$, welche der punktwweisen Fortsetzung von $b_{s,t}^{-1}$ entspricht. Enthält ein Baum r das Muster s , so enthält der Baum $\hat{B}_{s,t}(r)$ das Muster t . Enthält ein Baum v das Muster t , so folgt daraus, dass $\hat{B}_{s,t}^{-1}(v)$ das Muster s enthält.

Sei nun w ein beliebiger k -Baum, welcher das Muster s vermeidet. Wir wollen zeigen, dass $\hat{B}_{s,t}(w)$ das Muster t vermeidet. Würde dieser Baum das Muster t enthalten, so würde auch das Bild unter $\hat{B}_{s,t}^{-1}$ das Muster enthalten, ein Widerspruch.

□

Bemerkung 3.58. Eine weitere Eigenschaft dieser Wortdarstellung ist, dass jede Darstellung eines k -Baums auch als K -Baum, für $K \geq k$, interpretiert werden kann. Damit folgt, dass jede Bijektion aus dem obigen Lemma als Bijektion zwischen K -Bäumen interpretiert werden kann. Wodurch zusätzliche Wilf-Äquivalenzen gefunden werden können.

3.5 Unzusammenhängende Teilbäume

Nun, da wir die zusammenhängenden Substrukturen näher betrachtet haben, kommen wir zu den unzusammenhängenden. Wir betrachten im Folgenden wieder die Binärbäume \mathcal{T} und werden anschließend an die Definition, wann genau ein solches Muster enthalten wird, einige simple Beispiele kennen lernen. Als Basis dient [26].

Definition 3.59. Für zwei Binärbäume s und t gilt, dass s genau dann t als unzusammenhängendes Muster enthält, falls t , durch Kontraktionen⁴ von Kanten, aus s erzeugt werden kann.

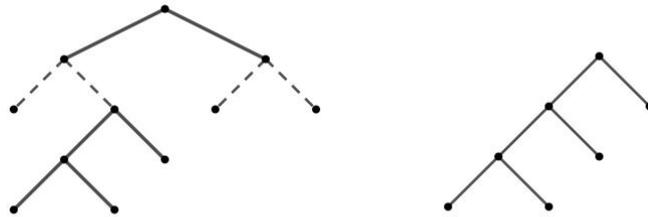


Abbildung 6

Aus der Definition folgt sofort, dass die Zahl A_n der Bäume, welche ein Muster t zusammenhängend vermeiden, größer sein muss als die Zahl a_n der Binärbäume, mit n internen Knoten, welche ein Muster t unzusammenhängend vermeiden. Damit ist auch die erzeugende Funktion

$$av_t(x) = \sum_{n \geq 0} a_n x^n \tag{34}$$

punktweise kleiner als $Av_t(x)$. Die Beispiele L , $L \wedge L$ und left-comb_2 werden wir hier auslassen, denn aufgrund ihrer Einfachheit entsprechen die dazugehörigen erzeugenden Funktionen den erzeugenden Funktionen für zusammenhängende Muster. Interessant wird es ab den Bäumen mit 3 internen Knoten, denn hier

⁴Kontraktion siehe Appendix 6.3

3 Bäume

zeichnen sich zum ersten Mal Unterschiede ab.

Wie wir im vorherigen Kapitel bereits festgestellt haben, gilt für die Bäume $t_1 = \text{left-comb}_3$ und $t_5 = \text{right-comb}_3$, $A_n = M_n$. Für die zweite Äquivalenzklasse t_2 , t_3 und t_4 gilt $A_n = 2^{n-1}$.

Wir beginnen mit dem Baum $t_1 = \text{left-comb}_3$, welcher aufgrund der Links-Rechts-Symmetrie in der gleichen Wilf-Äquivalenzklasse wie t_5 liegen muss.

Satz 3.60. *Es gilt*

$$\text{av}_{t_1}(x) = \text{av}_{t_5}(x) = \frac{x(1-x)}{1-2x} = \text{Av}_{t_2}(x).$$

Beweis. Es ist zu zeigen, dass $a_n = 2^{n-1}$ gilt. Ein Baum vermeidet in genau zwei Fällen left-comb_3 . Erstens: das linke Kind der Wurzel ist ein Blatt und der rechte Teilbaum vermeidet left-comb_3 . Hier haben wir a_{n-1} Möglichkeiten. Oder andererseits, der Baum hat die Form $a_L(i) \wedge a_R(i)$, wobei gelten muss, dass die Wurzel von a_L als linkes Kind ein Blatt hat und der rechte Teilbaum das Muster left-comb_2 vermeidet und $n-i-2$ interne Knoten hat. Für a_R gilt, dass er left-comb_3 vermeiden muss und i interne Knoten hat. Es gibt genau einen Baum, welcher left-comb_2 vermeidet, nämlich right-comb_i . Daraus folgt, dass es für diesen Fall $\sum_{i=0}^{n-2} a_i$ Möglichkeiten gibt. Damit gilt

$$a_n = a_{n-1} + \sum_{i=0}^{n-2} a_i = \sum_{i=0}^{n-1} a_i.$$

Aus den Anfangsbedingungen $a_0 = 1$ und $a_1 = 1$ erhalten wir das Gewünschte. □

Satz 3.61. *Alle Binärbäume mit drei internen Knoten liegen in der gleichen Wilf-Äquivalenzklasse und es gilt*

$$\text{av}_{t_i}(x) = \frac{x(1-x)}{1-2x} = \text{Av}_{t_2}(x).$$

Beweis. Wir müssen zeigen, dass auch t_2 und t_3 die gleiche erzeugende Funktion besitzen.

Zunächst betrachten wir den Baum t_2 . Dieser Fall funktioniert ganz analog zu t_1 . Es gibt wieder zwei Möglichkeiten. Erstens: der linke Knoten ist ein Blatt und der rechte Teilbaum ist ein t_2 vermeidender Baum der Größe $n-1$. Im zweiten Fall ist das linke Kind der Wurzel die Wurzel eines Teilbaums. Für diesen

3 Bäume

Teilbaum muss gelten, dass das rechte Kind der Wurzel ein Blatt ist und der linke Teilbaum das Muster right-comb_2 vermeidet. Aus der gleichen Berechnung folgt das gewünschte Ergebnis.

Für den Baum t_3 gilt Folgendes: Mindestens eines der beiden Kinder der Wurzel ist ein Blatt. Der andere Teilbaum muss wieder t_3 vermeiden, hat aber nur mehr a_{n-1} Möglichkeiten. Zusammen ergibt das $a_n = 2a_{n-1}$ und mit den üblichen Anfangsbedingungen erhalten wir wieder das Gewünschte.

□

Auf den ersten Blick erscheint es womöglich nicht wirklich interessant, dass es bei Binärbäumen mit drei internen Knoten nur eine Äquivalenzklasse gibt. Wie wir später sehen werden, ist dies der Vorbote des wichtigsten Satzes dieses Abschnitts.

Zunächst benötigen wir wieder die Menge aller Binärbäume, welche ein gegebenes Muster t (unzusammenhängend) vermeiden und das Muster p (zusammenhängend) an der Wurzel enthalten, $\mathcal{L}_t(p)$. Die zugehörige erzeugende Funktion wird mit $g_t^p(x)$ bezeichnet. Klarerweise gilt $\text{av}_t(x) = g_t^L(x)$. Natürlich gilt wieder eine zu Gleichung (31) entsprechende Aussage

$$g_t^L(x) = x + g_t^{L \wedge L}(x). \quad (35)$$

Es gibt für $t = t_l \wedge t_r$ auch das Pendant zu Gleichung (30)

$$g_t^{L \wedge L}(x) = g_{t_l}^L(x)g_{t_r}^L(x) + g_{t_l}^L(x)g_{t_r}^L(x) - g_{t_l}^L(x)g_{t_r}^L(x). \quad (36)$$

Ein Muster wird dann vermieden, wenn eine der beiden Seiten die eine Hälfte des Musters vermeidet. Der dritte Term muss aufgrund des Inklusions-Exklusions-Prinzips hinzugefügt werden, da sonst Bäume doppelt gezählt würden. Kombiniert man jetzt diese beiden Gleichungen, so erhält man

$$\text{av}_t(x) = x + \text{av}_{t_l}(x)\text{av}_t(x) + \text{av}_t(x)\text{av}_{t_r}(x) - \text{av}_{t_l}(x)\text{av}_{t_r}(x).$$

Damit erhalten wir

$$\text{av}_t(x) = \frac{x - \text{av}_{t_l}(x)\text{av}_{t_r}(x)}{1 - \text{av}_{t_l}(x) - \text{av}_{t_r}(x)}. \quad (37)$$

Beispiel 3.62. Wir wollen nun die Funktionen $\text{av}_{\text{left-comb}_n}(x)$ berechnen. Der Einfachheit halber werden wir die erzeugenden Funktionen der Bäume, welche left-comb_n vermeiden mit $\text{av}_n(x)$ bezeichnen.

Es gilt $\text{left-comb}_n = \text{left-comb}_{n-1} \wedge L$. Damit und mit Gleichung (37) erhalten wir

3 Bäume

$$\text{av}_n(x) = \frac{x - \text{av}_{n-1}(x)\text{av}_L(x)}{1 - \text{av}_{n-1}(x) - \text{av}_L(x)}. \quad (38)$$

Da es keine Bäume gibt, welche das Muster L vermeiden, gilt $\text{av}_L(x) = 0$. Deswegen ergibt sich

$$\text{av}_n(x) = \frac{x}{1 - \text{av}_{n-1}(x)}.$$

Durch vollständige Induktion könnte man zeigen, dass die folgende Funktion diese Gleichung erfüllt und auch mit den Anfangsbedingungen übereinstimmt. Aufgrund der Länge des Beweises werden wir das hier aber nicht durchführen, siehe [26, S.7]. Es gilt

$$\text{av}_n(x) = \frac{\sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} (-1)^i \binom{n-i-1}{i} x^{i+1}}{\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^i \binom{n-i}{i} x^i}.$$

Nun kommen wir zu der wohl interessantesten Aussage in diesem Abschnitt.

Lemma 3.63. *Für zwei Binärbäume mit n internen Knoten s und t gilt*

$$\text{av}_s(x) = \text{av}_t(x).$$

Beweis. Wir nehmen an, dass die Aussage für alle $k < n$ gilt. Wir betrachten die Bäume t^* , mit $k-1$ internen Knoten links von der Wurzel und s^* mit k internen Knoten links von der Wurzel. Aus Gleichung (38) folgt, dass

$$\begin{aligned} \text{av}_{t^*}(x) &= \frac{x - \text{av}_{k-1}(x)\text{av}_{n-k}(x)}{1 - \text{av}_{k-1}(x) - \text{av}_{n-k}(x)} \\ &= \frac{x - \text{av}_{k-1}(x) \frac{x}{1 - \text{av}_{n-k-1}(x)}}{1 - \text{av}_{k-1}(x) - \frac{x}{1 - \text{av}_{n-k-1}(x)}} \\ &= \frac{x(-1 + \text{av}_{k-1}(x) + \text{av}_{n-k-1}(x))}{1 - x - \text{av}_{k-1}(x) - \text{av}_{n-k-1}(x) + \text{av}_{k-1}(x)\text{av}_{n-k-1}(x)}. \end{aligned}$$

Für die zweite erzeugende Funktion gehen wir analog vor

$$\begin{aligned} \text{av}_{s^*}(x) &= \frac{x - \text{av}_k(x)\text{av}_{n-k-1}(x)}{1 - \text{av}_k(x) - \text{av}_{n-k-1}(x)} \\ &= \frac{x - \frac{x}{1 - \text{av}_{k-1}(x)}\text{av}_{n-k-1}(x)}{1 - \frac{x}{1 - \text{av}_{k-1}(x)} - \text{av}_{n-k-1}(x)} \\ &= \frac{x(-1 + \text{av}_{k-1}(x) + \text{av}_{n-k-1}(x))}{1 - x - \text{av}_{k-1}(x) - \text{av}_{n-k-1}(x) + \text{av}_{k-1}(x)\text{av}_{n-k-1}(x)}. \end{aligned}$$

3 Bäume

Damit gilt

$$\text{av}_{t^*}(x) = \text{av}_{s^*}(x),$$

woraus sofort folgt, dass die Aussage für alle Binärbäume mit n internen Knoten gilt.

□

Nun, da wir wissen, dass es für jedes n nur eine Wilf-Äquivalenzklasse innerhalb von \mathcal{T}_n gibt, wir auch dank Beispiel 3.62 in jeder dieser Klassen einen Repräsentanten kennen, gilt Folgendes.

Korollar 3.64. *Sei t ein beliebiger Binärbaum mit n internen Knoten, dann gilt*

$$\text{av}_t(x) = \text{av}_n(x) = \frac{\sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} (-1)^i \binom{n-i-1}{i} x^{i+1}}{\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^i \binom{n-i}{i} x^i}. \quad (39)$$

Nun können wir versuchen, unser erlangtes Wissen in anderen Bereichen zu nutzen. Dafür betrachten wir eine Bijektion ψ zwischen den Binärbäumen mit n internen Knoten \mathcal{T}_n und den Permutationen der Länge n , welche das Muster 132 vermeiden, $\mathcal{S}_n(132)$.

Definition 3.65. Die Abbildung $\psi : \mathcal{T}_n \rightarrow \mathcal{S}_n(132)$ ist folgendermaßen definiert. Für einen beliebigen Baum t markieren wir die internen Knoten mit verschiedenen Zahlen aus $\{1, \dots, n\}$. Die Wurzel bekommt die größte Markierung, der rechte Teilbaum mit k internen Knoten bekommt die k kleinsten noch freien Markierungen, der linke Teilbaum die restlichen. Wir fahren auf diese Weise rekursiv fort, bis jeder interne Knoten markiert wurde. Nun kontrahieren wir solange die Kanten zu Blättern, bis nur noch ein Knoten übrig ist, wobei, falls das linke Kind (resp. das Rechte) eines Knotens v kontrahiert wird, dessen Markierung links (resp. rechts) an die Markierung von v angehängt wird. Die Markierung des letzten Knotens entspricht dem Bild von t unter der Abbildung ψ .

Lemma 3.66. *Die oben definierte Abbildung ψ ist wohldefiniert und eine Bijektion.*

Beweis. Die Wohldefiniertheit ist klar, da der oben beschriebene Algorithmus für jeden Baum ein Ergebnis liefert. Da die Markierungen wie oben beschrieben vergeben werden, wird das Muster 132 vermieden. Die Injektivität ist klar und kann durch Induktion gezeigt werden. Da beide Mengen gleiche Größe besitzen muss es sich um eine Bijektion handeln. □

3 Bäume

Mit Hilfe der Bijektion ψ können wir nun einen Zusammenhang zu Permutationen feststellen. Sehr interessant ist auch, dass ψ Substrukturen von Binärbäumen in natürlicher Weise in Substrukturen der Permutationen überführt.

Satz 3.67. *Für einen beliebigen Binärbaum t mit $k \geq 1$ internen Knoten und a_n , die Zahl der Binärbäume mit n internen Knoten, welche das Muster t vermeiden, gilt für alle $n \geq 0$*

$$a_n = s_n(132, 12 \dots k). \quad (40)$$

Beweis. Für diesen Beweis benötigen wir die Bijektion ψ und erneut die Bäume left-comb_k . Das Bild von left-comb_k unter ψ ist die aufsteigende Permutation der Länge k . Nun müssen wir uns überlegen, dass eine Permutation aus $\mathcal{S}_n(132)$ genau dann eine aufsteigende Permutation der Länge k vermeidet, falls der zugehörige Baum left-comb_k vermeidet. Sei nun t ein Binärbaum, welcher left-comb_k enthält. Nun folgt aus der Definition von ψ , dass die aufsteigenden Markierungen des Teilbaums so angeordnet werden, dass die Permutation das Muster $12 \dots k$ enthält. Für Fall zwei nehmen wir an, dass unsere Permutation das Muster $12 \dots k$ enthält und überlegen uns, dass ψ^{-1} left-comb_k enthalten muss. Der Knoten mit der Markierung k muss in seinem linken nachfolgenden Teilbaum den Knoten mit der Markierung $k-1$ enthalten. Kontrahiert man alle bis auf die linke Kante vom Knoten k und setzt rekursiv fort, so erhält man left-comb_k . Da alle Binärbäume mit k internen Knoten in der gleichen Wilf-Äquivalenzklasse liegen, gilt das Gewünschte. □

Korollar 3.68. *Mit den gleichen Annahmen wie oben gilt*

$$a_n = s_n(132, k(k-1) \dots 21).$$

Beweis. Ersetzt man im obigen Beweis left-comb_k durch right-comb_k und aufsteigende durch absteigende Permutation, so erhält man das Gewünschte. □

3.6 Unbalancierte Teilbäume

Zu guter letzt beenden wir dieses Kapitel mit einem Teil über unbalancierte Teilbäume, insbesondere Bäume, welche die Raupeneigenschaft erfüllen. Grundlage dafür ist [12]. Es geht wieder um Binärbäume, jedoch werden wir wieder zusammenhängende Teilstrukturen untersuchen. Abgesehen vom Interesse aus biologischer Sicht, sind diese Art der Bäume für diese Arbeit interessant, da es

3 Bäume

eine Verbindung zu Mustervermeidung in Permutationen gibt. Desweiteren ist noch hervorzuheben, dass dieses Muster eine Besonderheit hat. Es kann nämlich nicht vermieden werden.

Definition 3.69. Ein Binärbaum t wird Raupe genannt, falls jeder Knoten in t entweder ein Blatt ist, oder ein Blatt als direkten Nachfolger besitzt.

Beispiele für Raupen sind die Bäume left-comb_k . Klarerweise enthält jeder Baum einen Teilbaum, welche die Raupeneigenschaft erfüllt. Von besonderem Interesse, vorallem für die Biologie, ist die Zahl der internen Knoten des größten Teilbaums mit der Raupeneigenschaft. Denn Bäume dieser Art werden meist durch die natürliche Selektion erzeugt. Der Name unbalancierte Teilbäume kommt daher, dass Raupen nach dem Colless-Index die Unbalanciertesten sind, siehe [12].

Definition 3.70. Für einen Baum t und Knoten i bezeichnet $t_l(i)$ (resp. $t_r(i)$) den linken (resp. rechten) nachfolgenden Teilbaum. Die Zahl $\Delta_t(i)$ ist definiert als der Betrag der Differenz der Größe der Teilbäume $t_l(i)$ und $t_r(i)$. Der Colless-Index ist definiert als

$$\frac{2}{(n-2)(n-1)} \sum_{i \text{ Knoten von } t} \Delta_t(i).$$

Dieser Wert liegt zwischen 0 und 1, wobei 0 den balancierten Bäumen und 1 den total unbalancierten entspricht.

Bemerkung 3.71. Für ein Baum t ist genau dann eine Raupe, falls der Colless-Index von t exakt 1 ist.

Definition 3.72. Sei t ein beliebiger binärer Baum, r der größte Teilbaum von t , welcher die Raupeneigenschaft erfüllt, so bezeichnet $\xi(t)$ die Anzahl der internen Knoten von r .

Klarerweise gilt $\xi(t) \geq 1$, was bedeutet, dass wir dieses Muster nicht vermeiden können. Stattdessen werden wir versuchen die Zahl der Binärbäume mit $\xi(t) = k$ zu bestimmen. Hierfür benötigen wir die erzeugende Funktion $F_k^-(x)$. Dies ist die erzeugende Funktion aller Bäume mit $\xi(t) \leq k$. Sie entspricht der Gleichung

$$F_k^-(x) = x + (F_k^-)^2 - 2^{k-1}x^{k+1},$$

da eine Raupe mit Größe kleiner oder gleich k entweder dem Baum L entspricht, oder wir konstruieren sie, indem wir an die Wurzel zwei Raupen mit Größe kleiner oder gleich k anhängen. Hier müssen wir jedoch die Fälle abziehen, in denen wir eine Raupe der Größe $k+1$ produzieren. Dies passiert falls wir eine

3 Bäume

Raupe der Größe k mit einem Blatt kombinieren. Aus dieser Gleichung erhalten wir die explizite erzeugende Funktion

$$F_k^-(x) = \frac{1 - \sqrt{1 - 4x + 2^{k+1}x^{k+1}}}{2}.$$

Desweiteren gilt

$$F_k(x) = F_k^-(x) - F_{k-1}^-(x).$$

Nun kommen wir noch zu einer weiteren Eigenschaft der Raupen. Hierfür benötigen wir wieder die Bijektion ψ .

Satz 3.73. *Unter der Bijektion ψ korrespondieren die Binärbäume, welche die Raupeneigenschaft erfüllen, mit den Permutationen, welche die Muster 132 und 231 vermeiden.*

Beweis. Sei t ein beliebiger Binärbaum. Wir müssen zeigen, dass t eine Raupe ist, genau dann wenn $\psi(t)$ die Permutation 231 enthält. Unter ψ bekommen die Knoten rekursiv Markierungen, indem die Wurzel die höchste freie Markierung bekommt, die rechten Nachfolger die kleinsten freien und die linken Nachfolger die größten freien. Dies entspricht genau dem Muster 231. Daraus lässt sich folgern, dass die Permutation das Muster 231 genau dann vermeidet, falls für jeden Knoten in t nur ein Nachfolger markiert wird, was genau den Raupen entspricht.

□

Wir haben gezeigt, dass Raupen mit n internen Knoten genau den Permutationen aus $\mathcal{S}_n(132, 231)$ entsprechen. Nun wollen wir noch weiter gehen und zeigen, dass Teilbäume eines Baumes genau dann Raupen sind, wenn die dem Teilbaum entsprechenden Markierungen das Muster 231 vermeidet.

Definition 3.74. Sei $\pi = \pi_1 \dots \pi_n$ eine Permutation, so ist das Bild der Funktion $r_\pi(\pi_i)$ definiert als jene Einträge π_k , für die

1. $\pi_k \leq \pi_i$ und
2. alle Einträge von π zwischen π_i und π_k sind kleiner oder gleich π_i

gilt.

Offensichtlich entspricht $r(\pi_i)$ der größten zusammenhängenden Teilmenge von π mit der Eigenschaft, dass π_i deren Maximum ist.

Satz 3.75. *Sei t ein beliebiger Binärbaum, so entspricht die Zahl $\xi(t)$, also die Zahl der internen Knoten der größten Raupe, der Länge der längsten Permutationen der Menge*

$$\{r_{\psi(t)}(\pi_i)\}_{i \leq n} \cap \mathcal{S}(231).$$

3 Bäume

Beweis. Sei v der Knoten, welcher unter ψ die Markierung m erhält und t_v jener Teilbaum mit Wurzel v . Man kann sich leicht überlegen, dass die Permutation $\psi(t_v)$ mit $r_{\psi(t)}(m)$ übereinstimmt. Mit dem obigen Satz ist der Baum t_v genau, dann eine Raupe, wenn diese Permutation das Muster 231 vermeidet. Nun ist $\xi(t)$ die Zahl der internen Knoten der größten dieser Teilbäume. Damit folgt das Gewünschte.

□

4 Catalan-Strukturen

Die Catalan-Zahlen C_n spielen in der Kombinatorik eine wichtige Rolle und kommen in verschiedensten Strukturen vor. Es ist also kein Wunder, dass auch innerhalb der Mustervermeidung die Catalan-Zahlen eine wesentliche Rolle spielen. Dies zeigt sich auch daran, dass in den vorherigen Kapiteln mehrmals die Catalan-Zahlen aufgetaucht sind.

Dieses Kapitel beginnen wir mit einer eher unbekannteren Struktur, den Catalan-Folgen. Die Mustervermeidung in diesem Bereich ähnelt zwar jener der Permutationen, jedoch gibt es doch gewisse Unterschiede. Anschließend werden wir die Catalan-Zahlen verallgemeinern und stoßen auf einen Satz, welcher mit der Mustervermeidung in diversen Bereichen zu tun hat. Sogar ein Pendant zu diesem Satz, mithilfe der verallgemeinerten Motzkin-Zahlen, wird auftreten.

4.1 Catalan-Folgen

Die erste Struktur, in der wir Mustervermeidung betreiben wollen, ist die der Catalan-Folgen. Die Aussagen dieses Unterkapitels stammen aus [15].

Definition 4.1. Eine Catalan-Folge der Länge n ist definiert als eine endliche Folge aus nicht negativen ganzen Zahlen $w = w_1 w_2 \dots w_n$, welche folgende zwei Bedingungen erfüllt

- i) $w_{i+1} \geq w_i - 1$, für alle $1 \leq i \leq n$
- ii) Für alle $k > 0$ gilt, dass für kleinste i , für das $w_i = k$ gilt, i_1 und i_2 existieren mit $i_1 \leq i \leq i_2$, sodass $w_{i_1} = w_{i_2} = k - 1$ gilt.

Punkt i) besagt, dass Catalan-Folgen nur 'langsam' fallen können. Punkt ii) besagt, dass sie nur begrenzt schnell wachsen können. Denn vor jeder Zahl müssen bereits alle kleineren Zahlen bis dorthin vorgekommen sein. Zusätzlich müssen nach jeder Zahl alle kleineren Zahlen noch einmal vorkommen.

Beispiel 4.2. Simple Catalan-Folgen sind für $n = 0, 1, 2, 3$:

$$\{0\}, \{00\}, \{000, 010\}, \{0000, 0100, 0110, 0010, 0101\}.$$

Lemma 4.3. Es existiert eine Bijektion ζ von den Catalan-Folgen der Länge n zu den Dyck-Pfaden der Länge $2n - 2$.

Den Beweis dazu gibt es in [14].

Muster, die Catalan-Folgen vermeiden beziehungsweise enthalten können, sind endliche Folgen aus nicht negativen ganzen Zahlen.

Definition 4.4. Die Abbildung $\text{red}(\cdot)$ (für Reduktion) ist definiert als jene Abbildung, welche innerhalb einer Folge von Zahlen alle Vorkommnisse der i -t größten Zahl durch die Zahl i ersetzt. Zusätzlich nennen wir zwei Folgen v, w ordnungsisomorph, falls $\text{red}(v) = \text{red}(w)$ gilt.

Definition 4.5. Eine Catalan-Folge w enthält ein Muster p genau dann, wenn es eine Teilfolge von w gibt, welche ordnungsisomorph zu p ist. Andernfalls vermeidet w das Muster p .

Vergleicht man die Kriterien der Mustervermeidung in Permutationen mit jener der Catalan-Folgen, so stellt man fest, dass in beiden unzusammenhängende Muster auftreten. Auch der zweite Punkt von Bemerkung 1.4 der Permutationen kann durch die Funktion $\text{red}(\cdot)$ interpretiert werden. Damit ist die Mustervermeidung in den Catalan-Folgen eine Verallgemeinerung zu jene der Permutationen.

4.1.1 Die konstante Folge der Länge k

Zunächst befassen wir uns mit dem Muster $p_k = 1 \dots 1$ (der Länge k). Wir bezeichnen die Menge aller Catalan-Folgen der Länge n , welche das Muster p vermeiden, mit $a_p(n)$ und die zugehörige erzeugende Funktion $A_p(x)$. Klarerweise wird das Muster $p_{1\dots 1}$ genau dann von einer Folge vermieden, wenn jede Zahl weniger als k mal darin vorkommt.

Wir starten mit einem interessanten Lemma über die erzeugende Funktion $A_p(x)$ für konstante Folgen, vgl. [15, S. 545, Theorem 2.1].

Lemma 4.6. Die erzeugende Funktion $A_p(x) = \sum_{n \geq 0} a_{1\dots 1}(n)x^n$ ist für alle k rational.

Beweis. Für diesen Beweis benötigen wir die Zahlen $a_{n,m}$. Diese sind definiert als die Zahl der Catalan-Folgen der Länge n , welche m 0er enthalten und das Muster p_k vermeiden. Klarerweise gilt $a_{n,m} = 0$ falls $m \geq k$ gilt, oder falls nicht $n \geq m \geq 1$ gilt. Es gilt $a_{n,1} = \delta_{n,1}$ und $a_{n,n} = \beta(n < k)$. Nun gilt, dass die Folgen, welche durch $a_{n,m}$ gezählt werden, zwischen einem und $k - 1$ 1er enthalten müssen. Diese Folgen kann man konstruieren, indem man vor eine Catalan-Folge der Länge $n - m$ eine null anhängt, jede Zahl der Folge um eins erhöht und anschließend noch $m - 1$ 0er in die $j + 1$ mögliche Positionen einfügt, also

$$\left(\binom{m+i-1}{i} - 1 \right) a_{n-m,j}$$

4 Catalan-Strukturen

Möglichkeiten. Damit gilt für $2 \leq m \leq k$ und $n \geq m$

$$a_{n,m} = \sum_{i=1}^{k-1} \left(\binom{m+i-1}{i} - 1 \right) a_{n-m,j}.$$

Für $2 \leq m \leq k$ genügen die erzeugenden Funktionen $A_m = \sum_{n \geq m} a_{n,m} x^n$ der Gleichung

$$A_m(x) = x^m + \sum_{j=1}^{k-1} \left(\binom{m+j-1}{j} - 1 \right) A_j(x).$$

Damit gilt für $\alpha_{i,j} = \binom{i+j}{i} - 1$ folgende Matrixdarstellung

$$\begin{pmatrix} 1 - x\alpha_{1,0} & -x\alpha_{2,0} & \cdots & -x\alpha_{k-1,0} \\ -x^2\alpha_{1,1} & 1 - x^2\alpha_{2,1} & \cdots & -x^2\alpha_{k-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ -x^{k-1}\alpha_{1,k-2} & x^{k-1}\alpha_{2,k-2} & \cdots & 1 - x^{k-1}\alpha_{k-1,k-2} \end{pmatrix} \begin{pmatrix} A_1(x) \\ A_2(x) \\ \vdots \\ A_{k-1}(x) \end{pmatrix} = \begin{pmatrix} x \\ x^2 \\ \vdots \\ x^{k-1} \end{pmatrix}$$

Offensichtlich gilt für $x = 0$, dass die Determinante dieser Matrix gleich 1 ist. Daraus folgt, dass sie invertierbar ist. Damit gibt es ein $\epsilon > 0$, sodass sie für $x \leq \epsilon$ invertierbar sein muss. Nach der Cramer'schen Regel müssen daher alle $A_i(x)$ rational sein. Die Summe von rationalen Funktionen ist wieder rational, daher gilt das Gewünschte. □

Beispiel 4.7. Es gibt genau eine Catalan-Folge, in welcher keine Zahl doppelt vorkommt. Für die entsprechende erzeugende Funktion gilt $A_{p_2}(x) = x$. Für die erzeugende Funktion für das Muster $p_3 = 111$ gilt, $A_{p_3}(x) = A_1(x) + A_2(x)$ (wie in der obigen Matrix). Es gilt $A_1(x) = x$. Für $A_2(x)$ gilt dann

$$\begin{aligned} A_2(x) &= x^2 + x^2 \left(\binom{2+1-1}{1} - 1 \right) x + x^2 \left(\binom{3}{2} - 1 \right) A_2(x) \\ A_2(x)(1 - 2x^2) &= x^2 + x^3 \end{aligned}$$

Damit gilt für das Muster $p_3 = 111$

$$A(x) = x + \frac{x^2 + x^3}{1 - 2x^2} = \frac{x(1 + x - x^2)}{1 - 2x^2}.$$

Als nächstes vereinfachen wir unsere Aufgabenstellung und beschäftigen uns mit dem Muster $p_3 = 111$ und versuchen die Koeffizienten $a_{111}(n)$ zu berechnen.

4 Catalan-Strukturen

Korollar 4.8. Für $n \geq 2$ gilt

$$a_{111}(n) = 2^{\lfloor \frac{n-2}{2} \rfloor}.$$

Beweis. Die Koeffizienten $a_{111}(n)$ entsprechen $[x^n]A_{p_3}(x)$. Damit gilt

$$[x^n] \frac{x(1+x-x^2)}{1-2x^2} = [x^{n-1}](1+x-x^2)(1+2x^2+4x^4 \dots (2x^2)^j \dots).$$

Sei n gerade, so können wir daraus schlussfolgern, dass $a_{111}(n) = 2^{\frac{n-2}{2}}$ gilt. Für ungerades n gilt

$$= [x^{n-1}](1-x^2) \sum_{j \geq 0} (2x^2)^j = [x^{n-1}]1 + \sum_{j \geq 1} 2^{j-1}x^{2j}.$$

Damit folgt was zu zeigen war. □

Die Autoren von [15] beschäftigten sich überdies noch mit allen Mustern der Länge drei und stellten fest, dass jedes der Muster eine rationale erzeugende Funktion hat. Zusätzlich berechneten sie die Koeffizienten $a_p(n)$ für alle diese Muster.

4.2 Modulare Catalan-Zahlen

In diesem Abschnitt kommen wir zu den modularen Catalan-Zahlen $C_{k,n}$. Diese wurden ursprünglich zur Verallgemeinerung der Assoziativität genutzt, kommen jedoch, genau wie die gewöhnlichen Catalan-Zahlen, in verschiedenen Bereichen vor. Außergewöhnlich ist jedoch, dass die modularen Catalan-Zahlen ungewöhnlich oft mit Themen der Mustervermeidung zusammenhängen. Klarerweise wird ihnen hier dafür ein Unterkapitel gewidmet. Die Inhalte stammen aus [16] von Nickolas Hein und Jia Huang.

4.2.1 k-Assoziativität

Der Ursprung der modularen Catalan-Zahlen liegt, wie bereits erwähnt in der Verallgemeinerung der Assoziativität. Für eine Menge X mit einer binären Operation $*$: $X \times X \rightarrow X$ ergibt der Ausdruck $x_0 * x_1 * \dots * x_n$ im Allgemeinen noch keinen Sinn. Erst mit Hilfe von Klammern lässt sich der Ausdruck interpretieren. Wir legen jedoch hiermit fest, dass, falls keine Klammern gesetzt sind, wir

4 Catalan-Strukturen

den Ausdruck von links nach rechts auswerten. Die Zahl der möglichen Klammersetzungen wird durch Catalan-Zahlen C_n gegeben und bildet dadurch eine obere Schranke im Sinne der Interpretationsmöglichkeiten. Sollte die Operation $*$ assoziativ sein, so gibt es nur eine einzige Möglichkeit, um den Ausdruck zu interpretieren. Dazwischen gibt es aber auch noch Abstufungen, denn die Operation $*$ wird k -assoziativ genannt, falls

$$x_0 * (x_1 * \cdots * x_{k+1}) = (x_0 * \cdots * x_k) * x_{k+1} \quad (41)$$

gilt.

Beispiel 4.9. Ein Beispiel für eine k -assoziative Operation kann auf jedem Ring R definiert werden. Sei $\omega \in R$ mit multiplikativer Ordnung k , dann wird durch $a * b := \omega a + b$ eine solche Operation definiert. Für $x_i \in R$ gilt

$$x_0 * (x_1 * \cdots * x_{k+1}) = \omega x_0 + \sum_{i=1}^{k+1} x_i \omega^{k-i+1} = (x_0 * \cdots * x_k) * x_{k+1}.$$

Zwei Klammersetzungen werden k -äquivalent genannt, falls sie im Sinne der k -Assoziativität äquivalent sind.

Definition 4.10. Die modularen Catalan-Zahlen $C_{k,n}$ sind definiert als die Zahl der k -Äquivalenzklassen von Klammersetzungen des Ausdrucks $x_0 * x_1 * \cdots * x_n$.

Die gewöhnliche Assoziativität entspricht der 1-Assoziativität, woraus sich ergibt, dass $C_{1,n} = 1$ gilt. Desweiteren gilt $C_{k,k} = C_k$.

4.2.2 Binärbäume

Wie bereits in Kapitel 3.3 erklärt, entspricht jeder Binärbaum einer Klammersetzung und vice versa. Als Konsequenz kann man die Struktur der k -Äquivalenz in natürlicher Weise auf Binärbäume übertragen.

Definition 4.11. Für einen Binärbaum t mit n internen Knoten, welcher am Knoten v den Teilbaum $s = (t_0 \wedge t_1 \wedge \cdots \wedge t_k) \wedge t_{k+1}$ besitzt, ist die Rechts- k -Rotation von t am Knoten v definiert als jener Baum, welcher t entspricht, aber an der Stelle v statt s den Baum $s' = t_0 \wedge (t_1 \wedge t_2 \wedge \cdots \wedge t_{k+1})$ besitzt. Die inverse Abbildung wird Links- k -Rotation am Knoten v genannt.

Definition 4.12. Wir definieren eine Ordnungsrelation, die k -assoziative Ordnung, auf der Menge der Binärbäume \mathcal{T}_n . Für zwei Bäume t und t' gilt $t \leq t'$ genau dann, wenn t durch endlich viele Links- k -Rotationen aus t' erzeugt werden kann.

Bemerkung 4.13. Die k -assoziative Ordnung ist eine Halbordnung, vgl. [16, S. 4].

Die Zusammenhangskomponenten dieser Halbordnung werden k -Komponenten genannt.

Proposition 4.14. Die Zahl der k -Komponenten auf \mathcal{T}_n entspricht $C_{k,n}$.

4.2.3 Ebene Bäume

Nun gehen wir weiter zu den ebenen Bäumen. Mithilfe der Bijektion γ aus Definition 3.43 können wir Binärbäume mit n internen Knoten in ebene Bäume mit $n+1$ Knoten überführen. Wir werden jedoch eine zusätzliche Eigenschaft der Abbildung kennen lernen und hier auch beweisen, dass es sich um eine Bijektion handelt.

Definition 4.15. Sei t ein Binärbaum aus \mathcal{T}_n . Für einen Knoten v aus t definieren wir die Linkstiefe von v als die Zahl der linken Kanten, welche zwischen v und der Wurzel liegen. Nun markieren wir die Blätter aufsteigend von links nach rechts mit $\{0, 1, \dots, n\}$. Mit $l_i(t)$ bezeichnen wir die Linkstiefe vom i -ten Blatt des Baumes t . Mit $l(t)$ bezeichnen wir den Vektor $(l_0(t), l_1(t), \dots, l_n(t))$.

Bemerkung 4.16. Das Bild eines Baumes mit auf diese Weise markierten Blättern, ist unter γ in sogenannter Preorder-Reihenfolge markiert.

Für ebene Bäume können wir eine ganz ähnliche Struktur definieren.

Definition 4.17. Sei T ein ebener Baum mit $n+1$ Knoten. Wir definieren den Gradvektor $d(T)$ als den Vektor $(d_0(T), d_1(T), \dots, d_n(T))$. Wobei $d_i(T)$ dem Grad des Knotens i des Baumes T , welcher in Preorder-Reihenfolge markierter wurde, entspricht.

Satz 4.18. Sei t ein beliebiger Binärbaum mit n internen Knoten und Linkstiefenvektor $l(t) = (l_0, l_1, \dots, l_n)$, so hat das Bild von t unter γ folgenden Gradvektor $d(\gamma(t)) = (d_0, d_1, \dots, d_n)$ und es gilt

$$l_i = d_0 + \dots + d_i - i. \quad (42)$$

Beweis. Nach Konstruktion von γ ist klar, dass $l_0 = d_0$ gilt. Nun betrachten wir zwei Fälle.

1.) $d_{i-1} = 0$: Damit gilt, dass der Knoten v_{i-1} von $\gamma(t)$ keine Kindknoten besitzt. Damit muss das Blatt $i-1$ von t ein Blatt sein und zusätzlich das rechte Kind eines Knotens. Nun betrachten wir den kürzesten Weg zum Blatt i . Dieser Weg entspricht einer Kante nach Nordwesten, danach eine nach Nordosten, danach

4 Catalan-Strukturen

eine nach Südosten und danach $p \geq 0$ nach Südwesten. Für die Linkstiefe gilt nun $\delta_i = \delta_{i-1} - 1 + p$. Nach der Anwendung von γ muss gelten, dass der Knoten v_i genau p Kindknoten hat, da für jeden Schritt nach Südwesten eines dazu kommt. Demnach gilt $\delta_i = \delta_{i-1} + d_i - 1$.

2.) $d_{i-1} \neq 0$: Nun muss das Blatt $i - 1$ das linke Kind eines Knotens sein. Der kürzeste Weg zum nächsten Blatt besteht aus einem nordöstlichen Schritt, einem südöstlichen Schritt und p südwestlichen Schritten. Wie gehabt erhalten wir $\delta_i = \delta_{i-1} - 1 + d_i$.

□

Proposition 4.19. *Sei T ein ebener Baum mit $n + 1$ Knoten. Für den Gradvektor $d(t) = (d_0, \dots, d_n)$ gilt Folgendes für alle i aus $\{0, \dots, n\}$*

$$d_0 + \dots + d_n = n, \quad d_0 + \dots + d_{i-1} \geq i. \quad (43)$$

Beweis. Da der Baum n Nichtwurzelknoten besitzt muss die Summe der d_i gleich n sein. Für $i \neq n$ ist die Linkstiefe des i -ten Blatts $l_i \geq 1$. Damit gilt

$$d_0 + \dots + d_i = l_i + i \geq i.$$

□

An dieser Stelle können wir den Beweis von Lemma 3.44 bringen. Für bessere Übersicht werden wir es hier noch einmal formulieren.

Lemma 4.20. ⁵ *Die Abbildung $\gamma: \mathcal{T}_n \rightarrow \mathcal{PT}_{n+1}$ ist eine Bijektion.*

Beweis. Nach Konstruktion ist klar, dass die Abbildung jedes Blatt eines Binärbaumes in einen Knoten eines ebenen Baumes überführt. Sei $\delta(t)$ der Linkstiefenvektor, so ist mit Gleichung (42) der Gradvektor des ebenen Baums eindeutig definiert, wobei obige Proposition gelten muss. Um zu zeigen, dass es sich bei der Abbildung γ um eine Bijektion handelt, reicht es zu zeigen, dass es ein Inverse zu γ gibt. Wir betrachten einen beliebigen ebenen Baum, Proposition 4.19 gelten muss. Gleichung (42) liefert uns daher einen eindeutigen Linkstiefenvektor, welcher den Binärbaum eindeutig codiert.

□

Nun können wir mithilfe der Linkskontraktion γ die k -assoziative Ordnung auf die ebenen Bäume übertragen.

Definition 4.21. Für zwei ebene Bäume s, t mit $n + 1$ Knoten gilt $s \leq t$ genau dann wenn für die Urbilder $\gamma^{-1}(s) \leq \gamma^{-1}(t)$ gilt.

⁵Die Abbildung γ wird auch Linkskontraktion genannt.

4 Catalan-Strukturen

Auch die k -Rotation hat ein Pendant in der Welt der ebenen Bäume.

Definition 4.22. Für einen ebenen Baum t ist der Up-(resp. Down)- k -slide am Knoten v definiert als die Abbildung γ^{-1} verkettet mit der Links- (Rechts) $-k$ -Rotation am Knoten $\gamma^{-1}(v)$, gefolgt von γ .

Diese Abbildung wird aus einem einfachen Grund Up- k -slide genannt. Sei in einem ebenen Baum t der Knoten v Kindknoten vom Knoten w und die ebenen Bäume T_1, \dots, T_k die k rechtesten Nachfolger vom Knoten v , so entspricht der Up- k -slide dem Transferieren der Nachfolgebäume T_i vom Knoten v zum Knoten w , wobei sie dort direkt rechts vom Knoten v plaziert werden. Einen Up- k -slide kann man also genau dann durchführen, falls es einen Nichtwurzelknoten v gibt, mit k oder mehr Kindknoten.

Korollar 4.23. *Durch einen Up- k -slide am Knoten v (zum Knoten w) ändert sich nichts an der Preorder-Reihenfolge der Knoten und für den Gradvektor ändern sich genau zwei Einträge*

$$d'_v = d_v - k, \quad d'_w = d_w + k. \quad (44)$$

Die inverse Abbildung dazu wird Down- k -slide genannt. Diese kann durchgeführt werden, falls es einen Knoten mit $k + 1$ oder mehr Kindknoten gibt. Sei v ein solcher Knoten mit den Kindern v_0, \dots, v_n . Hier besteht die Wahl an welchem Kindknoten v_i die Abbildung ausgeführt wird, mit der einzigen Bedingung, dass es muss mindestens k Kinder rechts davon geben muss. Nach dem Ausführen des Down- k -slides ist v_i der Elternknoten von $v_{i+1} \dots v_{i+k}$. Ansonsten wird die Struktur des Baums nicht verändert.

Korollar 4.24. *Ein Baum ist genau dann minimal bezüglich der k -assoziativen Ordnung (oder auch k -minimal), falls jeder Nichturzelknoten weniger als k Kindknoten besitzt. Ein Baum ist genau dann maximal bezüglich dieser Ordnung (oder auch k -maximal), wenn jeder Knoten weniger als $k + 1$ Kindknoten besitzt.*

Lemma 4.25. *Jede k -Komponente besitzt ein eindeutiges minimales Element.*

Beweis. Nach Korollar 4.23 sind zwei ebene Bäume höchstens dann in der gleichen k -Komponente, wenn ihr Gradvektor kongruent modulo k ist. Betrachten wir nun zwei verschiedene minimale Elemente, dann gilt für den Gradvektor, dass jeder der Einträge, außer der zur Wurzel korrespondierende, kleiner als k sein muss. Daraus folgt nun, dass die beiden Elemente in verschiedenen k -Komponenten liegen müssen.

□

4 Catalan-Strukturen

Damit gilt aber auch, dass die k -Komponenten der Binärbäume jeweils nur ein minimales Element besitzen. Einmal mehr kommen hier die Bäume left-comb_k ins Spiel. Nach Gleichung (42) ist das Bild von left-comb_k jener ebene Baum, welcher aus der Wurzel und ihren k Kindknoten besteht.

Satz 4.26. ⁶ *Ein Binärbaum mit n internen Knoten ist genau dann k -minimal, wenn er den Baum $\text{left-comb}_k^1 = L \wedge \text{left-comb}_k$ vermeidet. Ein Binärbaum ist genau dann k -maximal, wenn er left-comb_{k+1} vermeidet.*

Bevor wir den Beweis dieses Satzes durchführen, müssen wir uns aber noch kurz mit der Inversen zu γ beschäftigen. Diese Abbildung führt ebene Bäume in Binärbäume folgendermaßen über. Jeder Knoten wird auf den Baum left-comb_k abgebildet, wobei k dem Grad des Knotens entspricht. Mit den k Kindknoten wird rekursiv verfahren. Das Bild des i -ten Kinds ersetzt das $(i+1)$ -te Blatt von left-comb_k . Begonnen wird bei der Wurzel.

Beweis. Ein ebener Baum ist genau dann k -maximal, falls jeder Knoten weniger als $k+1$ Kinder besitzt, also der Grad jedes Knotens kleiner $k+1$ ist. Nach der Beschreibung von γ^{-1} muss das Bild eines k -maximalen Baums left-comb_{k+1} vermeiden.

Ein ebener Baum ist genau dann k -minimal, falls jeder Nichtwurzelknoten weniger als k Kindknoten besitzt, oder äquivalent dazu, der Nachfolger eines beliebigen Knotens muss einen Grad kleiner k besitzen. Das Bild unter γ^{-1} muss deshalb left-comb_k^1 vermeiden.

□

Definition 4.27. Die Menge aller Binärbäume mit n internen Knoten, welche left-comb_k^1 vermeiden, wird mit $\mathcal{T}_{k,n}$ bezeichnet.

Die Zahl der k -Komponenten auf \mathcal{T}_n entspricht $C_{k,n}$. Da jede Komponente genau ein minimales Element besitzt und die k -minimalen Elemente genau die Elemente aus $\mathcal{T}_{k,n}$ sind gilt

$$C_{k,n} = |\mathcal{T}_{k,n}|$$

4.2.4 Hauptsatz über die modularen Catalan-Zahlen

Der folgende Satz ist wohl der wichtigsten dieser Arbeit. Er verbindet in zentraler Weise verschiedene Bereiche der Mustervermeidung in verschiedenen kombinatorischen Strukturen.

Satz 4.28. *Für $n \geq 0$ und $k \geq 1$ werden folgende kombinatorische Strukturen durch die modularen Catalan-Zahlen $C_{k,n}$ gezählt.*

⁶Der Ausdruck \wedge wurde in Gleichung (3.3) definiert.

4 Catalan-Strukturen

1. *Permutationen der Länge n , welche das sowohl das Muster $1-3-2$ (zusammenhängend) als auch $23\dots(k+1)1$ (unzusammenhängend) vermeiden*
2. *Dyck-Pfade der Länge $2n$, welche das zusammenhängende Muster DU^k (wie in Def. 2.43) vermeiden*
3. *Binärbäume mit n internen Knoten, welche das Muster comb_k^1 (wie in Def. 3.41) vermeiden*
4. *Ebene Bäume mit $n+1$ internen Knoten, wobei jeder Nichtwurzelknoten weniger als k Kindknoten hat*
5. *Partitionen, welche durch $(n-1, n-2, \dots, 1, 0)$ beschränkt sind und in denen jede positive Zahl weniger als k Mal vorkommt*
6. *$2 \times n$ standard Young Tableaus, welche keine Abfolge von k aufeinander folgenden Zahlen in der oberen Reihe beinhalten, welche nicht $1, 2, \dots, l$ für $1 \leq l \leq n$ entsprechen*

Der Beweis für die Punkte 3. und 4. wurde bereits in den letzten beiden Unterkapiteln über Binärbäume beziehungsweise ebene Bäume erbracht. Den Beweis für den Punkt 1. funktioniert analog zum Beweis von Satz 4.33 Abschnitt Permutationen nur, dass wir comb_k^1 verwenden. Die Beweise für Punkte 5. und 6. werden wir hier nicht bringen, sie sind in Paper [16] auf Seite 11 und 12 zu finden.

Um den Beweis für Dyck-Pfade durchzuführen benötigen wir eine weitere Bijektion von den ebenen Bäumen zu den Dyck-Pfaden.

Definition 4.29. Die Abbildung $\bar{\epsilon} : \mathcal{PT}_{n+1} \rightarrow \mathcal{D}_n$ ist für einen ebenen Baum T mit Gradvektor $d(T) = (d_0, d_1, \dots, d_n)$ definiert als jener Dyck-Pfad, welcher dem Wort

$$U^{d_0} DU^{d_1} D \dots DU^{d_n}$$

entspricht.

Lemma 4.30. *Die Abbildung $\bar{\epsilon}$ ist wohldefiniert und eine Bijektion.*

Beweis. Nach Proposition 4.19 sind die zwei Bedingungen aus Bemerkung 2.2, dass es n Us und n Ds geben muss, und vor jedem i -ten D bereits i Us stehen müssen, erfüllt.

Für den zweiten Teil dieses Beweises konstruieren wir die Inverse zu dieser Abbildung. Für einen beliebigen n -Dyck-Pfad ist offensichtlich, wie wir zu dem Vektor $d = (d_0, d_1, \dots, d_n)$ kommen. Nun müssen wir zeigen, dass es für einen solchen Vektor einen zugehörigen ebenen Baum aus $n+1$ Knoten gibt. Sei $T_0 = L$ der

4 Catalan-Strukturen

Baum, welcher nur aus einem Knoten besteht. Wir markieren diesen einen Knoten. In jedem Schritt für $i = 1, \dots, n$ konstruieren wir einen neuen Baum T_i aus T_{i-1} , wobei wir an den zuletzt markierten Knoten d_{i-1} Kindknoten anhängen und dann den nächsten in Preorder-Reihenfolge noch nicht markierten Knoten markieren. Der Baum T_n ist unser gesuchter ebener Baum.

□

Beweis. (Satz 4.28 - Dyck-Pfade)

Mit der Bijektion \bar{e} können wir ebene Bäume mit $n + 1$ internen Knoten, wobei jeder Nichtwurzelknoten weniger als k Kindknoten hat, in n -Dyck-Pfade überführen. Für die Gradvektoren dieser Bäume muss gelten, dass jeder bis auf den ersten Eintrag kleiner als k ist. Damit gilt, dass das Bild eines solchen Baumes mit beliebig vielen Up-Steps beginnen darf, aber nach einem Down-Step keine k aufeinander folgende Up-Steps folgen dürfen. Das bedeutet, der Dyck-Pfad muss das zusammenhängende Muster DU^k vermeiden.

□

Nun, da wir unseren Teil über die modularen Catalan-Zahlen abgeschlossen haben können wir direkt ins nächste Unterkapitel springen. Genau wie die Catalan eng mit den Motzkin-Zahlen zusammenhängen, so sind die modularen Catalan-Zahlen direkt mit den verallgemeinerten Motzkin-Zahlen verbunden.

4.2.5 Die verallgemeinerten Motzkin-Zahlen

Definition 4.31. Die verallgemeinerten Motzkin-Zahlen $M_{k,n}$ sind definiert als die Anzahl der k -maximalen Elemente von \mathcal{T}_n .

Anhand der Binärbäume können wir sehen, dass für fixes n die verallgemeinerten Motzkin-Zahlen $M_{k,n}$ und die modularen Catalan-Zahlen $C_{k,n}$ verschachtelt sind.

Lemma 4.32. *Für fix es n gilt*

$$C_{1,n} \leq M_{1,n} \leq C_{2,n} \leq \dots \leq C_{n,n} \leq M_{n,n}.$$

Beweis. Jede k -Komponente hat ein eindeutiges k -minimales Element. Damit gilt $C_{k,n} \leq M_{k,n}$. Nun betrachten wir $\mathcal{T}_{k+1,n}$, also jene Binärbäume, welche das Muster left-comb_{k+1}^1 vermeiden. Im Gegensatz dazu betrachten wir die Binärbäume, welche left-comb_{k+1} vermeiden. Da das eine Muster ein Teilbaum des anderen ist, ist es echt schwieriger, das kleinere Muster zu vermeiden. Damit gilt das Gewünschte.

□

4 Catalan-Strukturen

Die verallgemeinerten Motzkin-Zahlen sind sehr mit den modularen Catalan-Zahlen verbunden, daher ist es einleuchtend, wenn wir Verbindungen zu ähnlichen Strukturen finden können.

Satz 4.33. *Für $n \geq 0$ und $k \geq 1$ entspricht die Mächtigkeit folgender kombinatorischer Strukturen den verallgemeinerten Motzkin-Zahlen $M_{k-1,n}$.*

1. *Permutationen der Länge n , welche die Muster $1-3-2$ (zusammenhängend) und $12\dots k$ (unzusammenhängend) vermeiden*
2. *Dyck-Pfade der Länge $2n$, welche das zusammenhängende Muster U^k (wie in Def. 2.43) vermeiden*
3. *Binärbäume mit n internen Knoten, welche das Muster left-comb_k (wie in Def. 3.41) vermeiden*
4. *Ebene Bäume mit $n+1$ Knoten, wobei jeder Knoten weniger als k Kindknoten hat*
5. *Partitionen, welche durch $(n-1, n-2, \dots, 1, 0)$ beschränkt sind und in denen jede Zahl weniger als k Mal vorkommt*
6. *$2 \times n$ standard Young Tableaus, in denen keine Abfolge von k aufeinander folgenden Zahlen in der oberen Reihe vorkommt*

Auch hier wurde der Beweis für die Punkte 3 und 4 bereits erbracht. Die Beweise für die Punkte 5 und 6 werden hier nicht geführt.⁷

Beweis. (Satz 4.33 - Permutationen)

Mit der Abbildung ψ stehen Binärbäume mit n internen Knoten und Permutationen der Länge n , welche das Muster 132 vermeiden, in Verbindung. Wir müssen nun zeigen, dass ein Binärbaum genau dann left-comb_k vermeidet, falls eine solche Permutation das aufsteigende zusammenhängende Muster $1-2-\dots-k$ vermeidet. Für die erste Richtung nehmen wir an, dass der Baum t das Muster left-comb_k enthält. Dann folgt, dass $\psi(t)$ ein unzusammenhängendes Muster der Länge k aus direkt aufeinander folgenden Zahlen beinhalten muss. Äquivalent dazu ist, dass die Inverse dieser Permutation ein zusammenhängendes Muster der Länge k aus nicht unbedingt aufeinander folgenden Zahlen beinhalten muss.

Beweis. (Satz 4.33 - Dyck-Pfade)

Wir zeigen, dass die Aussagen 5 und 2 äquivalent sind. Ein ebener Baum t mit $n+1$ internen Knoten, wobei jeder Knotengrad kleiner k ist, hat einen Gradvektor, wobei jeder Eintrag kleiner k sein muss. Damit gilt mit der Bijektion \bar{c} ,

⁷Für genaueres zu Partitionen oder Young Tableaus siehe [16] S. 11 und 12.

4 Catalan-Strukturen

dass der zu t korrespondierende Dyck-Pfad das zusammenhängende Muster U^k vermeiden muss.

□

4.2.6 Explizite Berechnung

Die explizite Berechnung der modularen Catalan-Zahlen $C_{k,n}$ gestaltet sich überaus mühsam. Der Beweis ist in [16] ab Seite 13 zu finden. Beweisidee ist das Aufstellen von Funktionalgleichungen für die erzeugenden Funktionen

$$M_k(z) = \sum_{n \geq 0} M_{k,n} z^{n+1}, \text{ und } C_k(z) = \sum_{n \geq 0} C_{k,n} z^{n+1}.$$

Für $k \geq 0$ erhalten wir die Gleichung

$$M_k(z) = z + zM_k(z) + \cdots + zM_k^k(z).$$

Für $k \geq 1$ erhalten wir die Gleichungen

$$\begin{aligned} C_k(z) &= z/(1 - M_{k-1}(z)) \text{ und} \\ (C_k(z) - z)^k - C_k(z)^k + C_k(z)^{k-1} - zC_k(z)^{k-2} &= 0. \end{aligned}$$

Aufgrund dieser Berechnungen konnten die Autoren Nickolas Hein und Jia Huang dann folgenden Satz formulieren.

Satz 4.34. *Für $k \geq 1$ und $n \geq 0$ gilt*

$$M_{k-1,n} = \frac{1}{n+1} \sum_{0 \leq j \leq n/k} (-1)^j \binom{n+1}{j} \binom{2n-jk}{n},$$

und für $n, k \geq 1$ gilt

$$C_{k,n} = \sum_{0 \leq j \leq (n-1)/k} \frac{(-1)^j}{n} \binom{n}{j} \binom{2n-jk}{n+1}.$$

5 Verschiedenes

Nun, da wir die Schwerpunkte der Mustervermeidung abgehandelt haben, kommen wir zum Kapitel Verschiedenes. Denn es muss klargestellt werden, dass in den verschiedensten kombinatorischen Strukturen Mustervermeidung betrieben werden kann. Abgesehen von den Permutationen bilden die letzten drei Kapitel über die Dyck-Pfade, die diverse Arten von Bäumen und die Catalan-Strukturen die Schwerpunkte der Mustervermeidung vom heutigen Stand. Es gibt jedoch weitere wichtige und interessante Themen, die in dieser Arbeit nicht fehlen dürfen.

5.1 Strings

Wir beginnen mit den zusammenhängenden Strings, da sie vor allem in der Informatik, aber auch in anderen Bereichen zahlreiche Anwendungen finden. Wir haben jedoch im Kapitel der Gitterpfade bereits eine vollständige Theorie erarbeitet welche nicht nur die zusammenhängende Mustervermeidung der Strings beinhaltet, sondern, sie auch noch verallgemeinert. Wir werden deshalb nur schematisch die wichtigsten Punkte präsentieren.

Definition 5.1. Sei S ein beliebiges Alphabet, dann ist ein String $s = s_1 \dots s_n$ definiert als endliche Abfolge von Elementen von S .

Bemerkung 5.2. Strings der Länge n über dem Alphabet S werden mit S^n bezeichnet.

Definition 5.3. Ein String $s = s_1 \dots s_n$ über einem Alphabet S , enthält einen String $p = p_1 \dots p_k$ genau dann, falls es einen Index $1 \leq j \leq n$ gibt, sodass $s_{j+i} = p_i$ für alle $i \in \{1, \dots, k\}$ gilt.

Es werden hier also zusammenhängende Muster untersucht und wir möchten bemerken, dass die unzusammenhängenden Muster nicht durch unsere Arbeit in den Gitterpfaden abgedeckt wird und theoretisch noch zu untersuchen sind.

Mit unserer Arbeit über Gitterpfade könnten wir sogar Mustervermeidung in einem gewichteten Alphabet betreiben. Wir benötigen hierfür das charakteristische Polynom des Alphabets $P(w)$, das Autokorrelationspolynom des zu vermeidenden Musters p , bezeichnet mit $R(x, w)$ und den Kern $K(x, w)$ wie in Kapitel 2.6.

Satz 5.4. Für die erzeugende Funktion der Strings, welche das Muster p vermeiden, gilt

$$W(x, w) = \frac{R(x, w)}{K(x, w)}.$$

wobei die Variable x die Länge der Strings und w das Gewicht beschreibt. Im ungewichteten Fall können wir $w = 1$ setzen.

Für den ungewichteten Fall ist zu bemerken, dass die erzeugende Funktion nur von der Länge des Musters p und vom zugehörigen Autokorrelationspolynom abhängt.

5.2 Verallgemeinerung der Mustervermeidung

Wir setzen mit einem außerordentlich Interessanten Abschnitt fort. Die von Jarosław Grytczuk beschriebene Mustervermeidung auf Graphen ([17]) steht in starkem Kontrast zu der Mustervermeidung, welche im Bereich der Permutationen entwickelt wurde.

5.2.1 Variable Muster

Der erste Punkt dieses Papers bezieht sich auf die Verallgemeinerung der Non-repetitiveness für Strings.

Definition 5.5. Sei S ein Alphabet, dann wird der String $s = s_1 \dots s_n$ non-repetitiv genannt, falls es keine zusammenhängende Teilfolge der Form $yy = y_1 \dots y_k y_1 \dots y_k$ gibt.

Klarerweise wird diese Eigenschaft erst bei Strings der Länge zwei oder mehr interessant. Ein weiterer wichtiger Faktor ist die Größe des Alphabets, welches betrachtet wird. Man kann sich leicht überlegen, dass es nur begrenzt lange sich nicht wiederholende Strings über einem Alphabet mit Mächtigkeit eins oder zwei geben kann. Nun stellt sich die Frage ob es bei einem endlichen Alphabet überhaupt möglich ist einen beliebig langen nicht repetitiven String zu konstruieren. Eine Antwort darauf erhält man durch einen Satz, welcher von A. Thue bereits vor gut 100 Jahren aufgestellt wurde.

Satz 5.6. Für jedes Alphabet S mit $|S| \geq 3$ existiert ein beliebig langer non-repetitiver String.

Der Beweis dazu ist in [22] zu finden.

Die Theorie über sich nicht wiederholende Folgen ist für sich genommen schon interessant und reichhaltig, kann aber doch noch erweitert werden.

Definition 5.7. Sei X eine unendliche Menge an Variablen und $p = p_1 \dots p_k$ ein String über X . Ein String $s = s_1 \dots s_n$ enthält das variable Muster p , genau dann wenn s in nicht-leere Blöcke $B_1 \dots B_k$ unterteilt werden kann, sodass $B_i = B_j$ gilt, falls $p_i = p_j$ gilt.

Die Zerteilung in Blöcke ist ein vollkommen neuer Ansatz und sollte sicherlich auch in anderen Strukturen untersucht werden.

5.2.2 Mustervermeidung auf Graphen

Der zweite Punkt von [17] bezieht sich auf die Mustervermeidung unter dem Teilaspekt, dass Graphen mit eingebunden werden. Wir betrachten beliebige Graphen, wobei die Knoten mit Elementen aus S markiert werden.

Definition 5.8. Sei P ein einfacher Pfad auf einem markierten Graphen, dann wird die Folge der Markierungen, welche P durchläuft, mit η_P bezeichnet.

Definition 5.9. Ein markierter Graph enthält ein Muster w über S genau dann, wenn es einen einfachen Pfad P gibt und der zugehörige String η_P das Muster w enthält. Ansonsten sprechen wir davon, dass der Graph das Muster w vermeidet oder w -frei ist.

Bemerkung 5.10. Die obige Definition ist eine Verallgemeinerung der Mustervermeidung auf Strings, wobei es nicht relevant ist ob es sich um zusammenhängende Muster, unzusammenhängende Muster oder variable Muster handelt.

Die erste Fragestellung, die in diesem Kontext auftritt, ist, wie groß die Mächtigkeit des Alphabets sein muss, um überhaupt eine p -freie Markierung eines Graphen zuzulassen.

Definition 5.11. Für einen Graphen G , ein Muster p ist die Musterchromatische Zahl $\pi_p(G)$ definiert als die minimale Anzahl an verschiedenen Elementen, welche ein Alphabet enthalten muss, damit G p -frei markiert werden kann.

Leider gibt es in diesem Bereich noch nicht viele Ergebnisse. Dafür gibt es zahlreiche Vermutungen und die Hoffnung, dass noch einiges in dieser Richtung erforscht werden kann.

5.3 Mengenpartitionen

Im dritten Teil unseres Kapitels kommen wir zu den Mengenpartitionen basierend auf den Aussagen von [18]. Wir werden uns auf die Menge $\{1, \dots, n\}$ beschränken.

Definition 5.12. Eine geordnete Partition einer Menge M ist definiert als eine Folge von geordneten nichtleeren Blöcken, deren Vereinigung die Menge M ergibt.

5 Verschiedenes

Dieser Abschnitt beschäftigt sich mit den geordneten Partitionen der Menge $\{1, \dots, n\}$ in k verschiedene Blöcke, welche wir mit $\mathcal{OP}_{n,k}$ bezeichnen werden. Klarerweise entspricht die Menge $\mathcal{OP}_{n,n}$ den Permutationen der Länge n . Die Definition der Mustervermeidung verläuft sehr ähnlich zu der der Permutationen. Sie stellt aber in gewisser Hinsicht eine Verallgemeinerung dar.

Definition 5.13. Sei $B = B_1 \dots B_k$ eine Partition aus $\mathcal{OP}_{n,k}$. Dann enthält B die Permutation $p = p_1 \dots p_m$ genau dann, wenn es Indizes $1 \leq i_1 \leq \dots \leq i_m \leq k$ gibt, sodass b_j in B_{i_j} enthalten sind, wobei die Folge $b_1 \dots b_m$ ordnungsisomorph zu p sein muss. Andernfalls wird die Permutation vermieden.

Anders formuliert: Wenn es eine unzusammenhängende Teilfolge an Blöcken gibt, welche je ein Element enthalten, woraus sich eine zu p ordnungsisomorphe Folge ergibt.

Die geordneten Partitionen der Menge $\{1, \dots, n\}$ in k Blöcke, welche das Muster p vermeiden wird mit $\mathcal{OP}_{n,k}(p)$ bezeichnet und die Mächtigkeit dieser Mengen wird mit $\text{op}_{n,k}(p)$ bezeichnet. Klarerweise gilt

$$\text{op}_{n,n}(p) = s_n(p).$$

Es besteht also die Hoffnung, dass wir über die Mengenpartitionen einen neuen Zugang zur Mustervermeidung in den Permutationen erhalten. Jedoch gestaltet sich das Berechnen von $\text{op}_{n,k}(p)$ als überaus schwierig, für die nicht-trivialen Fälle.

An dieser Stelle kommt eine neue Bijektion θ ins Spiel.

Definition 5.14. Die Menge aller Strings über dem Alphabet $\{1, \dots, k\}$, wobei jede Zahl mindestens einmal vorkommen muss, wird mit $\mathcal{SW}_{n,k}$ bezeichnet. Die Menge aller oben beschriebenen Strings, welche zusätzlich das Muster p vermeiden wird mit $\mathcal{SW}_{n,k}(p)$ bezeichnet.

Definition 5.15. Die Abbildung $\theta : \mathcal{OP}_{n,k} \rightarrow \mathcal{SW}_{n,k}$ ist definiert als jene Abbildung, welche der Partition $B = B_1 \dots B_k$ das Wort $\theta(B) = w_1 \dots w_n$ zuordnet, wobei gilt, dass $w_i = j$ genau dann gilt, wenn $i \in B_j$ gilt.

Lemma 5.16. *Die oben beschriebene Abbildung θ ist wohldefiniert und eine Bijektion.*

Beweis. Wir beginnen mit der Wohldefiniertheit. Es wird jedem w_i eine eindeutige Zahl zugeordnet, da jedes i in genau einem Block vorkommt. Da jeder Block nichtleer ist, muss auch jede der Zahlen j im String vorkommen.

Nun zeigen wir, dass es sich um eine Bijektion handelt indem wir zeigen, dass es eine Inverse θ^{-1} gibt. Diese Abbildung ordnet jedem String $s = s_1 \dots s_n$ eine

5 Verschiedenes

geordnete Partition $P = P_1 \dots P_k$ der Menge $\{1, \dots, n\}$ in k Blöcke zu. Wobei $i \in P_j$ gilt, genau dann wenn $s_i = j$ gilt.

□

Die Bijektion θ ist besonders schön, da sie in natürlicher Weise die Muster der beiden Strukturen erhält, vgl [18].

Satz 5.17. *Eine geordnete Partition B aus $\mathcal{OP}_{n,k}$ enthält eine Permutation p genau dann, wenn das Wort $\theta(B)$ in $\mathcal{SW}_{n,k}$ das Muster p^{-1} enthält.*

Korollar 5.18. *Aus dem obigen Satz folgt für $n \geq 1$, $k \geq 1$ und eine feste Permutation p*

$$op_{n,k}(p) = |\mathcal{SW}_{n,k}(p^{-1})|.$$

Aus diesem Korollar folgt wiederum folgende Proposition.

Proposition 5.19. *Wir bezeichnen mit $[j]$ die Menge $\{1, \dots, j\}$ und mit $A^n(p)$ die Menge aller Strings der Länge n über dem Alphabet A , welche das Muster p vermeiden. Für eine Permutation p und $n, k \geq 1$ gilt*

$$op_{n,k}(p) = \sum_{j=1}^k \binom{k}{j} (-1)^{k-j} |[j]^n(p^{-1})|.$$

Beweis. Mit dem obigen Korollar und dem Inklusions-Exklusions-Prinzip folgt

$$|\mathcal{SW}_{n,k}(p^{-1})| = \sum_{A \subseteq [k]} (-1)^{k-|A|} |A^n(p^{-1})|.$$

Nun gibt es $\binom{k}{j}$ Möglichkeiten Teilmengen der Größe j auszuwählen. Für jede dieser Teilmengen mit $|A| = j$ gilt offensichtlich $|A^n(p^{-1})| = |[j]^n(p^{-1})|$. Damit erhalten wir das Gewünschte.

□

Diese von Anisse Kasraoui [18, S. 3, Proposition 1.1] aufgestellte Äquivalenz liefert uns einen Zugang, um die Zahlen $op_{n,k}(p)$ explizit zu berechnen. Dies ist ein wesentlicher Schritt, da, wie oben bereits bemerkt, die Zahlen $op_{n,n}(s)$ dem Term $s_n(p)$ entsprechen und damit von Bedeutung dafür sein könnten, in der Mustervermeidung der Permutationen voranzukommen.

6 Anwendungen

Schlussendlich sind wir im Teil über Anwendungen angelangt. Für rein mathematische Themen ist dies ja nicht immer so einfach. Doch obwohl die Mustervermeidung an sich als eher abstrakter gilt, gibt es doch direkte Anwendungen, zum Beispiel in der Informatik, der Biologie oder bei der Logistik Firma Amazon.

6.1 Informatik

In der Informatik wird die Mustervermeidung in der Form von pattern - matching verwendet, denn beim 'common subexpression problem' oder zu deutsch 'Problem der gemeinsamen Teilausdrücke' treten (binäre) Bäume auf, welche möglichst komprimiert werden sollen. Ziel ist, es Teilbäume nur einmal vorkommen zu lassen und jedes weitere Vorkommen durch einen Pointer zum ersten Vorkommen zu ersetzen. Bei dieser Prozedur wird aus dem Baum ein "dag" ("directed acyclic digraph") und es kann mitunter einiges an Arbeit, Speicher etc. eingespart werden, vgl. [19].

6.2 Biologie

Im Kapitel über Binärbäume, genauer gesagt im Unterkapitel Unbalancierte Teilbäume, wurde bereits erwähnt, dass Bäume, welche die Raupeneigenschaft erfüllen, von Interesse für die Biologie sind. Desweiteren findet die Mustervermeidung auch in einem anderen Teil der Biologie Anwendung, der Genetik. Hier ist vor allem die Mustervermeidung in Strings von Bedeutung, beispielsweise bei der Untersuchung der RNA. Diese besteht aus kurzen essenziellen Stücken, auch Module genannt, welche voneinander durch Sequenzen aus zufälligen Basenpaaren getrennt sind. Die Aufgabe ist nun, wichtige und unwichtige Sequenzen voneinander zu unterscheiden, was durch eine Kombination aus Statistik und Mustervermeidung realisiert wird, vgl. [20].

6.3 Praktische Anwendung

Die Informationen dieses Unterkapitels stammen aus [21].

Es gibt tatsächlich auch praktische Anwendungen für die Mustervermeidung. Bei der Firma Amazon etwa werden Roboter zum Warentransport innerhalb eines Lagers verwendet. Die Roboter müssen pro Stück, das in ein Paket gepackt wird, vier Arbeitsschritte bewältigen. Der erste Schritt (die Ware nehmen) und der letzte Schritt (die Ware abliefern), sind eindeutig. Die Schritte zwei und drei jedoch können auch vertauscht werden. Aus diesem Grund können die einzelnen Arbeitsschritte durch einen Diamanten dargestellt werden.

6 Anwendungen

Definition 6.1. Ein Diamant mit n Knoten ist definiert als Graph mit n Knoten, wobei es zwei ausgezeichnete Knoten 0 und 1 gibt. Die 0 bildet die erste Ebene. In der zweiten Ebene befinden sich die restlichen $n - 2$ Knoten, alle verbunden zur 0 und in der obersten Ebene befindet sich die 1, welche zu allen Knoten in der mittleren Ebene verbunden ist. Die Menge $\mathcal{E}_{n,d}$ bezeichnet die geordnete Menge von d Diamanten mit n Knoten, wobei alle Knoten auf folgende Weise mit Markierungen aus der Menge $\{1, \dots, nd\}$ markiert sind. Jeder Knoten bekommt eine eindeutige Markierung, wobei für jeden Diamanten gelten muss, dass die 1 die größte Markierung und die 0 die kleinste Markierung erhalten muss.

Das Verpacken eines Pakets mit d Waren kann durch ein Element aus $\mathcal{E}_{4,d}$ beschrieben werden. Seien nun ohne Beschränkung der Allgemeinheit die Waren ihrem Gewicht nach geordnet. Das bedeutet, der erste Diamant entspricht dem schwersten Paket und der letzte Diamant entspricht dem leichtesten. Hier tritt nun ein Problem auf, denn die Markierung der 1 entspricht dem Zeitpunkt, zu dem die Waren abgeliefert und ins Paket gegeben werden.

Beispiel 6.2. Hier sehen wir ein Element aus $\mathcal{E}_{4,3}$.

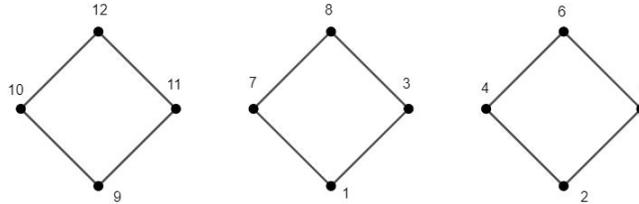


Abbildung 7

Es gilt, dass das leichteste Paket zum Zeitpunkt 6 abgelegt wird, darauf liegt das mittlere Paket mit Markierung 8 und zu guter letzt kommt das schwerste Paket zum Zeitpunkt 12.

Für einen Diamanten d gibt es eine zugehörige Permutation π_d . Diese entspricht der Markierung der 0, gefolgt von den Markierungen der mittleren Ebene, gelesen von links nach rechts, abgeschlossen durch die Markierung der 1. Für ein Element E aus $\mathcal{E}_{n,d}$ gibt es eine korrespondierende Permutation π_E . Diese entspricht der Permutation $\pi_1\pi_2 \dots \pi_d$, wobei die Permutation π_i die zum i -ten Diamanten zugehörige Permutation bezeichnet.

An dieser Stelle kommt die Mustervermeidung ins Spiel, denn das Problem, das im obigen Beispiel illustriert wird, kann dadurch vermieden werden, dass die entsprechende Permutation das Muster 321 vermeidet.

6 Anwendungen

Der Einfachheit halber werden wir ab hier mit \mathcal{E}_d die Menge $\mathcal{E}_{4,d}$ bezeichnen. Solche Diamanten, welche Muster aus der Menge P vermeiden, werden mit $\mathcal{E}_d(P)$ bezeichnet. An dieser Stelle wollen wir bemerken, dass für ein Element E aus $\mathcal{E}_d(P)$ und mit zugehöriger Permutation π_E , das Revers - Komplement π_E^{rc} wieder eine Permutation eines Elements aus $\mathcal{E}_d(P)$ ist.

Proposition 6.3. *Sei $\mathcal{E}_d(P)^{rc} := \{E : \pi_E = \pi^{rc} \wedge \pi \text{ ist die zugehörige Permutation eines Elements aus } \mathcal{E}_d(P)\}$, so gilt*

$$\mathcal{E}_d(P)^{rc} = \mathcal{E}_d(P^{rc}). \quad (45)$$

Aus dieser Proposition folgt, dass zwei Permutationen Wilf - Äquivalent sind, falls sie revers - komplementär zueinander sind. Für Permutationen der Länge drei gilt damit: $132 \sim_W 213$ und $231 \sim_W 312$.

Wir starten mit der Fragestellung, wie viele Elemente die Menge \mathcal{E}_d hat.

Satz 6.4. *Für $n \geq 4$ und $d \geq 1$ gilt*

$$|\mathcal{E}_{n,d}| = \frac{(vd)!}{n^d(n-1)^d}.$$

Beweis. Für einen Diamanten mit n Knoten sind zwei Knoten eindeutig zu markieren. Die restlichen $n-2$ Knoten können in beliebiger Weise markiert werden. Es gibt also $(n-2)!$ Möglichkeiten. Für d Diamanten gibt es demnach

$$\binom{nd}{d, \dots, d} (n-2)!^d = \frac{(nd)!((n-2)!)^d}{(n!)^d} = \frac{(nd)!}{n^d(n-1)^d}$$

Möglichkeiten.

□

Nun, da wir einige grundlegende Dinge geklärt haben, beschäftigen wir uns mit den Mustern der Länge drei, beginnend mit dem leichtesten.

Korollar 6.5. *Für das Muster $p = 123$ gilt*

$$|\mathcal{E}_{n,d}(p)| = 0.$$

Beweis. Die Permutation jedes Diamanten muss klarerweise das Muster 123 enthalten. Damit gibt es kein zulässiges Element, welches dieses Muster vermeidet.

□

6 Anwendungen

Für das nächste Muster fanden die Autoren [21] eine natürliche Bijektion von der Menge $\mathcal{E}_d(132)$ zu den Gitterpfaden wie sie im Appendix 6.3 beschrieben werden. Diese Bijektion konnte einige Eigenschaften der Permutationen, welche das Muster 132 vermeiden, erhalten, nämlich die rechts - links Maxima, die Abstiege und die Länge der längsten aufsteigenden Teilfolge. Daraus resultiert die folgende Äquivalenz.

Korollar 6.6.

$$|\mathcal{E}_{n,d}(132)| = |\mathcal{E}_{n,d}(213)| = \frac{\binom{d(n+1)}{d}}{nd + 1}$$

Das nächste Muster, das wir betrachten wollen, entspricht der Permutation 231. Dank Proposition 6.3 gelten alle Ergebnisse gleichzeitig auch für das Muster 312. Die Autoren von [21] konnten hier eine Rekursion aufstellen. Für die Zahlen $a_{n,j}^k$ gilt Folgendes.

Satz 6.7.

$$\mathcal{E}_{n,d}(231) = \mathcal{E}_{n,d}(312) = a_{n,n}^d = a_{n,n-1}^d + \sum_{i=1}^{d-1} a_{n,n-1}^i a_{n,n}^{d-i}$$

mit den Startwerten

$$a_{n,j}^1 = \begin{cases} 1, & \text{falls } j = 1 \\ C_{j-1}, & \text{falls } j = 2, \dots, n-1 \\ C_{n-2}, & \text{falls } j = n. \end{cases}$$

Damit sind fünf der sechs Permutationen der Länge drei abgehandelt. In [21] wurden auch noch jene Fälle untersucht, in denen zwei oder mehrere Permutationen der Länge drei vermieden werden. Das Muster 321 stellt sich als das schwerste heraus und leider waren die Autoren nicht in der Lage, es zu lösen. Damit bleibt es offen und ist repräsentativ dafür, dass es im Gebiet der Mustervermeidung noch viel zu erforschen gibt.

Appendix

Fine-Zahlen

Die Fine-Zahlen sind eng verwandt mit den Catalan-Zahlen. Die Informationen dieses Abschnitts und noch weitere sind in [24] zu finden.

Definition 6.8. Die Fine-Zahlen f_n sind definiert als Koeffizienten der erzeugenden Funktion $F(x) = \sum_n f_n x^n$. Es gilt

$$F(x) = \frac{1}{1 - x^2 C(x)^2} = \frac{C(x)}{1 + C(x)}, \quad (46)$$

wobei

$$C(x) = \frac{1 - \sqrt{1 - 4x}}{2x}$$

gilt.

Lemma 6.9. *Es gilt*

$$\lim_{n \rightarrow \infty} \frac{f_n}{C_n} = \frac{4}{9}. \quad (47)$$

Ballot-Zahlen

Die Definition der Ballot-Zahlen und auch deren explizite Formel stammt aus [25].

Definition 6.10. Die Ballot-Zahlen $b_{n,k}$ sind durch folgende Rekursion definiert. Für $n > 0$ und $k \geq 0$ gilt

$$b_{n,k} = \sum_{j=0}^k b_{n-1,j}. \quad (48)$$

Mit den Startwerten $b_{0,0} = 1$ und für $k > 0$ gilt $b_{0,k} = 0$.

Lemma 6.11. *Die explizite Formel für die Ballot-Zahlen entspricht*

$$b_{n,k} = \frac{n - k + 1}{n + 1} \binom{n + k}{n}$$

für $0 \leq k \leq n$.

Allgemeine Gitterpfade

Wir wollen noch eine weitere allgemeinere Definition von Gitterpfaden, oder auch Nord-Ost-Pfaden angeben.

Definition 6.12. Ein Nord-Ost-Pfad ist definiert als ein Gitterpfad von $(0, 0)$ nach (n, kn) mit Schritten $N = (0, 1)$ und $E = (1, 0)$, welche schwach unter der Geraden $y = kx$ liegen.

Bemerkung 6.13. Für das entsprechende Wort aus $\{N, E\}$ gilt, dass vor jedem i -ten N -Step bereits $k * i$ E -Steps gemacht worden sein müssen.

Lemma 6.14. Die Zahl solcher Pfade entspricht

$$\frac{1}{kn + 1} \binom{(k + 1)n}{n}.$$

Bemerkung 6.15. Für $k = 1$ entsprechen diese Pfade den Dyck-Pfaden, welche durch eine einfache Drehspiegelung erhalten werden können.

Schröder-Zahlen

Die Definition der Schröder Pfade passt nicht ganz zu der aus dem Kapitel über Gitterpfade, darum wollen wir sie hier kurz extra anführen. Wir betrachten die Exursionen mit Schritten aus dem Alphabet $S = \{(1, 3), (2, 2), (1, -1)\}$ von $(0, 0)$ nach $(4n, 0)$. Wobei zu beachten ist, dass der Eintrag $(2, 2)$ einem Schritt um zwei Einheiten entlang der x-Achse und zwei Einheiten entlang der y-Achse entspricht.

Definition 6.16. Die Schröder-Zahlen S_n beschreiben die Anzahl der Schröder-Pfade der Länge n .

Bemerkung 6.17. Schröder Pfade können auch als Nord-Ost-Pfade mit den Schritten $S = \{(0, 1), (1, 1), (1, 0)\}$ interpretiert werden.

Lemma 6.18. Es gilt

$$S_n = \frac{1}{kn + 1} \sum_{j=0}^n \binom{kn + 1}{n - j} \binom{kn + j}{j}.$$

Kontraktion

Sei $G = (V, E)$ ein Graph mit Knotenmenge V und Kantenmenge E , zwei Knoten v und w durch eine Kante e mit einander verbunden, dann entspricht für den Graphen G die Kontraktion der Kante e , $G' = (V', E')$. Wobei $V' = V \setminus v$ gilt und die Menge E' der Menge E ohne der Kante e entspricht, wobei zusätzlich jede Kante, welche den Knoten v mit einem anderen Knoten s verbinden würde, stattdessen den Knoten w mit s verbindet.

Literatur

- [1] Christian Krattenthaler, Permutations with restricted patterns and dyck paths, *Adv. Appl. Math.* 27 (2001), 510-530, Vienna, Austria, 2001.
- [2] Paul Peart and Wen-Jin Woan, Dyck paths with no peaks at height k , *Journal of Integer Sequences*, Vol. 4 (2001), Article 01.1.3, 6 pp., Washington, D.C. 20059, USA, 2001.
- [3] Toufik Mansour, Counting peaks at height k in a dyck path, *Journal of Integer Sequences* 5, Article 02..1.1, 33405 Talence Cedex, France, 2002.
- [4] Sen-Peng Eu, Shu-Chung Liu, and Yeong-Nan Yeh, Dyck Paths with peaks avoiding or restricted to a given set, *Studies in Applied Mathematics* 111:453–465, Kaohsiung, Taiwan, 2003.
- [5] A. Bernini, L. Ferrari, R. Pinzani, and J. West. Pattern-avoiding Dyck paths. *Discrete Math. Theor. Comput. Sci. Proc. AS*:683–694, 2013.
- [6] Andrei Asinowski, Axel Bacher, Cyril Banderier, and Bernhard Gittenberger. Analytic combinatorics of lattice paths with forbidden patterns: enumerative aspects. In S. Klein, C. Martín-Vide, and D. Shapira, editors, *Language and Automata Theory and Applications. LATA 2018*, volume 10782 of *Lecture Notes in Computer Science*, pages 195–206. Springer, 2018.
- [7] Andrei Asinowski, Axel Bacher, Cyril Banderier, and Bernhard Gittenberger, Analytic combinatorics of lattice paths with forbidden patterns: Asymptotic aspects and borges’s theorem. *LIPICs*, Vol. 110 – Aofa 2018, pages 10:1–10:14, 2018.
- [8] David Bevan, Derek Levin, Peter Nugent, Jay Pantone, Lara Pudwell, Manda Riehl, ML Tlachac, Pattern avoidance in forests of binary shrubs, *Discrete Mathematics and Theoretical Computer Science*, Vol. 18 no. 2, *Permutation Patterns 2015*, *Permutation Patterns* (July 21, 2016) dmtcs:1541, 22 pp., 2016.
- [9] Vladimir Dotsenko, Pattern avoidance in labelled trees, *Séminaire Lotharingien de Combinatoire*, B67b (2012), 27 pp., Luxembourg, 2012.
- [10] Derek Levin, Lara K. Pudwell, Manda Riehl, and Andrew Sandberg. Pattern avoidance in k -ary heaps. *Australas. J. Combin.*, 64:120–139, 2016
- [11] Eric S. Rowland, Pattern avoidance in binary trees, *Journal of Combinatorial Theory, Series A* 117 (2010) 741-758, New Orleans, LA 70118, USA, 2010.

Literatur

- [12] Filippo Disanto, Unbalanced subtrees in binary rooted ordered and un-ordered trees, Séminaire Lotharingien de Combinatoire 68 (2013), Article B68b, 14 pp., 2013.
- [13] Nathan Gabriel, Katherine Peske, Lara Pudwell, Samuel Tay, Pattern avoidance in ternary trees, Journal of Integer Sequences, Vol. 15 (2012), 19 pp., 2012.
- [14] Christian Stump, On a new collection of words in the catalan family, Journal of Integer Sequences, Vol. 17 (2014), Article 14.7.1, 5 pp., 2014.
- [15] Toufik Mansour, Mark Shattuck, Avoidance of classical patterns by Catalan sequences. Filomat. 8. 543-. 10.2298/FIL1703543M., 2017.
- [16] [1] N. Hein and J. Huang, Modular Catalan numbers, Euro. J. Comb. 61 (2017), 197–218 pp., 2017.
- [17] Jarosław Grytczuk, Pattern avoidance on graphs, Discrete Math. 307 (11–12) (2007) 1341–1346 pp., Poland, 2007.
- [18] Anisse Kasraoui, Pattern avoidance in ordered set partitions and words, Advances in Applied Mathematics, Volume 61, 85-101 pp., 2014.
- [19] P. Flajolet, P. Sipala, and J.-M. Steyaert. Analytic variations on the common subexpression problem. In Automata, Languages and Programming, pages 220-234. Springer-Verlag, 1990.
- [20] Manuel Lladser, M. Betterton, Rob Knight, Multiple pattern matching: A markov chain approach, Lladser, M. et al. (2008) Multiple pattern matching: a Markov chain approach. J. Math., Biol., 56, 51–92 pp., 2008.
- [21] Mitchell Paukner, Lucy Pepin, Manda Piehl, Jarred Wieser, Pattern avoidance in task-precedence posets, Discrete Mathematics and Theoretical Computer Science, Vol. 18 no. 2, Eau Claire, WI, USA, 2016.
- [22] A. Thue, Über unendliche Zeichenreihen, Norske Vid Selsk. Skr. I. Mat. Nat. Kl. Christiana, 7, Pages 1-22 (1906)
- [23] Emeric Deutsch, Dyck path enumeration, Discrete Mathematics, Volume 204, Issues 1–3, 6 June 1999, Pages 167-202, 1999.
- [24] Emeric Deutsch, Louis Shapiro, A survey of the Fine numbers, Discrete Mathematics, Volume 241, Issues 1–3, Pages 241-265, 28 October 2001.

Literatur

- [25] L. Carlitz, Sequences, paths, ballot numbers, *Fibonacci Quart.* 10 (1972), 531-549, 1972.
- [26] M. Dairyko, L. Pudwell, S. Tyner, and C. Wynn. Non-contiguous pattern avoidance in binary trees. *Electron. J. Combin* 19(3):P22, 21 pp., 2012.
- [27] Alois Panholzer.
http://info.tuwien.ac.at/panholzer/DM_Skriptum_V1.pdf
Skriptum, Diskrete Mathematik.