Colorings

Gregor Hink
Mat.-Nr. 227506

Seminar: Extremal Combinatorics
LuFG Theoretische Informatik, RWTH Aachen
Sommersemester 2004

Inhaltsverzeichnis

1 Einleitung

Die Färbbarkeit von Mengen hat in der Informatik eine wesentliche Bedeutung. Am stärksten untersucht wurde bisher die Färbbarkeit von Graphen, was darin begründet ist, dass sich viele Probleme in der Informatik auf Graphen übertragen lassen. Als Beispiel sei die Färbung von planaren Graphen genannt, der die Färbung von Landkarten zu Grunde liegt. Hier konnte gezeigt werden, dass vier Farben ausreichend sind. Ein weiteres Beispiel ist die NP-Vollständigkeit der 3-Färbbarkeit von Graphen, die ein wesentlicher Bestandteil der Theorie der NP-Vollständigen Probleme ist.

In dieser Ausarbeitung werden im nächsten Kapitel zuerst einige zentrale Begriffe definiert, welche die Grundlage für die folgenden Kapitel darstellen. Kapitel 3 zeigt an einigen Sätzen auf, unter welchen Bedingungen Mengen 2-färbbar sind. Den Hauptteil der Ausarbeitung bildet Kapitel 4, in welchem die Laufzeit von Färbungsalgorithmen näher betrachtet wird.

2 Grundlagen

In diesem Kapitel werden einige wichtige Begriffe definiert, die für den Rest der Ausarbeitung vorausgesetzt werden: Es sei X eine Menge von Punkten. Diese Menge ist unterteilt in verschiedene, nicht notwendigerweise disjunkte, Teilmengen. Die Menge dieser Teilmengen wird mit F bezeichnet.

Weiterhin sei $\{1, \ldots, r\}$ mit $r \geq 2$ eine Menge von Farben. Eine r-Färbung von X ist nun eine Abbildung $c : X \to \{1, \ldots, r\}$. Abbildung 1 zeigt eine solche Menge X von Punkten mit ihren Teilmengen, sowie eine mögliche 2-Färbung. Die beiden Farben sind hier als o bzw. als + dargestellt.

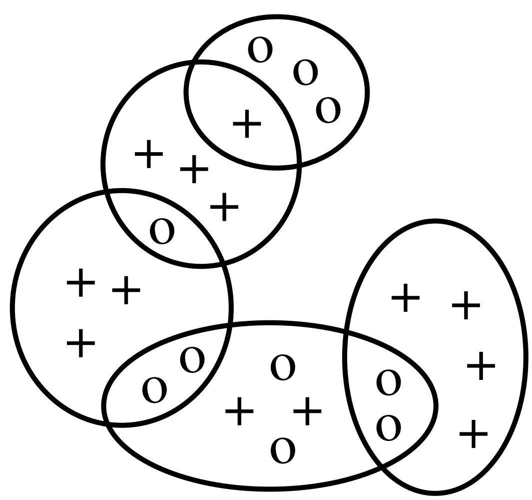

Abbildung 1: Eine mögliche 2-Färbung einer Menge X

Eine Teilmenge A von X heißt genau dann monochromatisch, wenn $c(a) = i$ für alle Elemente $a \in A$ und ein festes $i \in \{1, \ldots, r\}$ gilt. Eine Färbung von F ist genau dann gültig, wenn kein Element von F monochromatisch ist. Zu einer Menge F von Teilmengen ist die chromatische Zahl $\chi(F)$ definiert, als die kleinste Zahl von Farben, für die eine gültige Färbung für F existiert.

Wenn eine Menge F 2-färbbar ist, dann sagt man zu Ehren des deutschen Mathematikers Felix Bernstein (1878 – 1956) auch, sie hat die Eigenschaft B.

3 2-Färbbarkeit (Eigenschaft B)

In diesem Kapitel geht es darum zu zeigen, dass eine Menge unter bestimmten Voraussetzungen 2-färbbar ist.

Satz 1 *Wenn $|A \cap B| \neq 1$ für jeweils zwei Mengen $A \neq B \in F$ ist, dann ist F 2-färbbar.*

Beweis: Der Beweis geht über vollständige Induktion über die Mächtigkeit von X. Dabei sei $X = \{x_1, \ldots, x_n\}$. Als Induktionsanfang kann x_1 beliebig gefärbt werden.

Im Induktionsschritt seien nun $x_1, \ldots, x_i$ bereits gefärbt. Nun versucht man x_{i+1} mit der Farbe 0 zu färben. Ist dies nicht möglich, so existiert eine Teilmenge $A \subseteq X$ mit $x_{i+1} \in A$ und alle Punkte in $A \backslash \{x_{i+1}\}$ sind bereits mit Farbe 0 gefärbt. Nun bleibt noch die Möglichkeit x_{i+1} mit Farbe 1 zu färben. Ist auch dies nicht möglich, so existiert zusätzlich eine Teilmenge $B \subseteq X$ mit $x_{i+1} \in B$ und alle Punkte in $B \backslash \{x_{i+1}\}$ sind bereits mit Farbe 1 gefärbt.

Daraus folgt, dass der Durchschnitt $A \cap B = \{x_{i+1}\}$ ist. Dies ist aber ein Widerspruch zur Voraussetzung $|A \cap B| \neq 1$. Daher ist es möglich x_{i+1} zu färben.

□

Abbildung 2 zeigt eine solche Färbung für drei Mengen, deren paarweiser Schnitt jeweils die Mächtigkeit 2 hat oder die leere Menge ist.

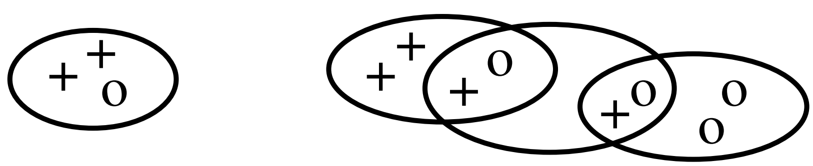

Abbildung 2: Beispiel zu Satz 1

Satz 2 *Sei F eine Menge von Teilmengen, die jeweils mindestens die Mächtigkeit k haben. Wenn $|F| < 2^{k-1}$ ist, dann ist F 2-färbbar.*

Beweis: Es seien $|X| = n$ und $A \in F$. Nun bezeichne $C(A)$ die Anzahl der Färbungen, bei denen A vollständig mit einer Farbe gefärbt ist. Da sich A dabei mit 2 Farben belegen lässt und für die übrigen $n - |A|$ Elemente aus X noch $2^{n-|A|}$ Kombinationen existieren folgt, dass $C(A) = 2 \cdot 2^{n-|A|}$ ist. Da jede Teilmenge von F mindestens die Mächtigkeit k hat, lässt sich $C(A)$ nach oben abschätzen durch

$$C(A) = 2 \cdot 2^{n-|A|} \leq 2 \cdot 2^{n-k} = 2^{n-k+1}.$$

Summiert man nun $C(A)$ für alle möglichen $A \in F$, so ergibt sich die maximale Anzahl der ungültigen Färbungen für F. Dies ergibt:

$$\sum_{A \in F} C(A) = \sum_{A \in F} 2^{n-k+1} = |F| \cdot 2^{n-k+1}$$

Da nach der Voraussetzung $|F| < 2^{k-1}$ ist, folgt unmittelbar: $|F| \cdot 2^{n-k+1} < 2^{k-1} \cdot 2^{n-k+1} = 2^n$. Da 2^n die Anzahl aller Färbungen für F darstellt und die maximale Anzahl der ungültigen Färbungen echt kleiner ist als die Anzahl der Färbungen insgesamt, muss also mindestens eine gültige Färbung für X existieren.

$\square$

Auch wenn eine Menge nicht immer 2-färbbar ist, so lassen sich unter bestimmten Bedingungen noch immer Teile der Menge gültig färben. Eine Menge F von Teilmengen ist genau dann k-einheitlich, wenn jedes Element von F die Mächtigkeit k hat.

Satz 3 *Für jede k-einheitliche Menge F von Teilmengen existiert eine 2-Färbung, bei der höchstens $|F| \cdot 2^{1-k}$ Elemente von F monochromatisch sind.*

Beweis: Es sei $|X| = n$ und $S \subseteq X$. Dann bezeichnet $M(S)$ die Anzahl der monochromatischen Elemente von F bei der Färbung, die alle Punkte von S mit der einen Farbe und alle anderen Punkte mit der anderen Farbe färbt. Berechnet man nun die Summe dieser monochromatischen Elemente über alle Teilmengen S so ergibt sich

$$\sum_{S \subseteq X} M(S) = \sum_{A \in F} 2 \cdot |\{S : A \subseteq S\}|.$$

Nun folgt, da jedes Element von F genau k Elemente besitzt, dass

$$\sum_{A \in F} 2 \cdot |\{S : A \subseteq S\}| = 2 \cdot |F| \cdot 2^{n-k} = |F| \cdot 2^{n-k+1}$$

ist. Es existiert daher mindestens eine Menge $S \subseteq X$ mit $M(S) \leq |F| \cdot 2^{1-k}$.

$\square$

4 Komplexität von Färbungsalgorithmen

Aus Sicht der Informatik sind die Fragestellungen aus Kapitel 3 zwar nicht unwichtig, wesentlich interessanter ist allerdings die Frage nach der Komplexität eines Färbungsalgorithmus. Aus der Komplexitätstheorie ist bekannt, dass die 3-Färbbarkeit von Graphen NP-Vollständig ist. Wie sieht es aber nun für 2 Farben aus? Eine erste Strategie wäre es, alle möglichen Kombinationen durchzugehen. Bei 2 Farben und n Punkten ergeben sich 2^n mögliche Kombinationen, so dass dieser erste Ansatz auch zu einer exponentiellen Laufzeit führt. Es stellt sich nun die Frage, ob es nicht auch schneller geht. Wie der Rest dieses Kapitels am Beispiel der Färbung von Kuben zeigen wird, ist dies im Allgemeinen nicht möglich.

Ein n-Kubus $\{0,1\}^n$ besteht aus 2^n Vektoren. Zu einem solchen Kubus ist ein Unterkubus $C \subseteq \{0,1\}^n$, bei dem einige Komponenten mit 0 bzw. 1 vorbelegt sind. Nicht vorbelegte Komponenten werden dabei mit einem $*$ bezeichnet, so dass sich ein solcher Unterkubus als Tupel über $\{0,1,*\}^n$ schreiben lässt. Ein Vektor von einem solchen Kubus ist genau dann gerade bzw. ungerade, wenn er eine gerade bzw. ungerade Anzahl an Einsen enthält. Ein n-Kubus ist genau dann gültig gefärbt, wenn jeder gerade Vektor weiß und jeder ungerade Vektor schwarz gefärbt ist.

Beispiel: Abbildung 3 zeigt eine graphische Darstellung des 3-Kubus $\{0,1\}^3$. Ein möglicher Unterkubus hierzu wäre die Vorderseite des Würfels, bestehend aus den vier Vektoren $(0,0,0),(0,0,1),(1,0,0)$ und $(1,0,1)$. Ein 4-Kubus $\{0,1\}^4$ ist die Menge der Vektoren $\{(0,0,0,0),(0,0,0,1),(0,0,1,0),(0,0,1,1),(0,1,0,0),(0,1,0,1),(0,1,1,0),(0,1,1,1),(1,0,0,0),(1,0,0,1),(1,0,1,0),(1,0,1,1),(1,1,0,0),(1,1,0,1),(1,1,1,0),(1,1,1,1)\}$. Ein möglicher Unterkubus dazu wäre $C = (0,*,1,1)$, bestehend aus den Vektoren $\{(0,0,1,1),(0,1,1,1)\}$. Ein gerader Vektor aus diesem Kubus ist zum Beispiel der Vektor $(0,1,1,0)$, ein ungerader Vektor $(0,1,1,1)$.

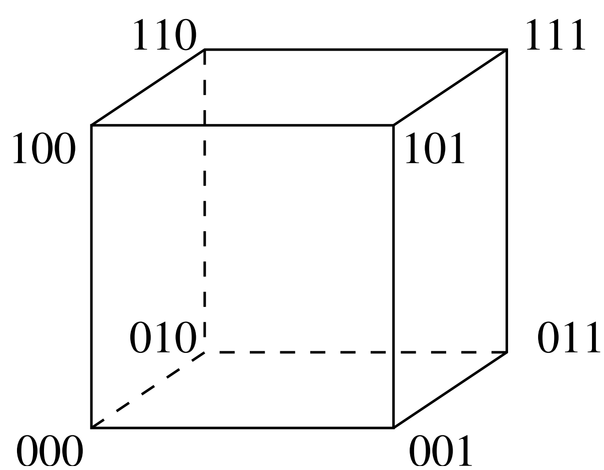

Abbildung 3: Graphische Darstellung des 3-Kubus

Im Folgenden wird nun von einem Algorithmus ausgegangen, der in jedem einzelnen Schritt einen Unterkubus weiß oder schwarz färbt. Das Gerüst für einen solchen Algorithmus sieht wie folgt aus:

1. Zu Beginn ist kein Vektor gefärbt.
2. Wähle in jedem Schritt einen Unterkubus und färbe alle seine Vektoren entweder weiß oder schwarz.
3. Einmal gefärbte Vektoren behalten ihre Farbe, auch wenn ein zugehöriger Unterkubus in einem späteren Schritt gefärbt wird.

Beispiel: (Fortsetzung) Eine mögliche Färbung des 4-Kubus mit dem obigen Algorithmus sähe dann folgendermaßen aus:

1. Färbe $(0,0,0,0), (0,0,1,1), (1,1,0,0)$ und $(1,1,1,1)$ weiß.
2. Färbe $(*,*,0,0), (*,*,1,1), (0,0,*,*)$ und $(1,1,*,*)$ schwarz, die in Teil 1 gefärbten Vektoren bleiben dabei weiß.
3. Färbe $(*,*,*,*)$ weiß, die in Teil 2 gefärbten Vektoren bleiben dabei schwarz.

Wie viele Schritte werden nun im Allgemeinen benötigt? Eine erste Strategie ist es, jeden geraden Vektor einzeln weiß und danach den gesamten Kubus schwarz zu färben. Da bereits vorher gefärbte Vektoren ihre Farbe behalten, färbt diese Strategie alle geraden Vektoren weiß und alle ungeraden Vektoren schwarz.

Da in einem solchen Kubus 2^{n-1} gerade Vektoren existieren, benötigt die Färbung der geraden Vektoren 2^{n-1} Schritte. Die Färbung der ungeraden Vektoren benötigt dagegen nur einen Schritt, da diese durch Färbung des gesamten Kubus in einem Schritt geschieht. Die Laufzeit des Algorithmus für diese Strategie ist also $2^{n-1}+1$. Dies ist auch wieder eine exponentielle Laufzeit, und es stellt sich die Frage, ob ein besserer Algorithmus existiert.

Satz 4 *Jeder Algorithmus, der einen n-Kubus färbt, benötigt mindestens $2 \cdot (\frac{3}{2})^{n-1}$ Schritte.*

Beweis: Es sei S_n die minimale Anzahl der benötigten Schritte. Nun lässt sich $S_n \geq 2 \cdot (\frac{3}{2})^{n-1}$ umformen zur Rekursion:

$$S_n \geq \frac{3}{2} S_{n-1} \text{ mit } S_1 = 2$$

Für den Fall $n = 1$ gilt offensichtlich $S_1 = 2$, indem man den geraden Vektor weiß und danach den ungeraden Vektor schwarz färbt, wozu zwei Schritte notwendig sind. Nun ist noch die Rekursion $S_n \geq \frac{3}{2} S_{n-1}$ zu zeigen.

Es sei P der Algorithmus mit der geringsten Laufzeit, der den n-Kubus korrekt färbt. P lässt sich nun als Folge $(C_1, a_1), (C_2, a_2), \ldots, (C_s, a_s)$ schreiben, was bedeutet, dass im i-ten Schritt die Vektoren des Unterkubus C_i mit der Farbe a_i gefärbt werden. Es gilt nun, dass $S_n = |P|$ die Anzahl der Schritte s ist.

Nun wird die erste Komponente eines Unterkubus C_i betrachtet, diese ist entweder 0 oder 1 oder $*$. Der Unterkubus C_i lässt sich daher als $0C_i'$ bzw. $1C_i'$ bzw. $*C_i'$ schreiben. Es sei P_0 bzw. P_1 der Teil des Algorithmus, welcher die i Schritte enthält, bei denen die erste Komponente von C_i die 0 bzw. die 1 ist. Analog sei P_0^* bzw. P_1^* der Teil des Algorithmus, welcher die i Schritte enthält, bei denen die erste Komponente von C_i die 0 oder $*$ bzw. die 1 oder $*$ ist. Die Länge $|P|$ des Algorithmus lässt sich nun aufteilen in

$$|P| = |P_0| + |P_1^*| = |P_1| + |P_0^*|.$$

Die Teile P_0^* und P_1^* des Algorithmus lassen sich so modifizieren, dass sie den $(n-1)$-Kubus färben. Für P_0^* geschieht dies durch Weglassen der ersten Komponente von jedem Unterkubus, bei P_1^* müssen zusätzlich die beiden Farben vertauscht werden.

Die Anzahl der Schritte lässt sich nun abschätzen zu $S_n = |P| \geq S_{n-1} + max(|P_0|, |P_1|)$. Nun muss also nur noch gezeigt werden, dass $max(|P_0|, |P_1|) \geq \frac{1}{2} \cdot S_{n-1}$ ist. Diese Aussage lässt sich umformen und es reicht dann noch zu zeigen:

$$|P_0| + |P_1| \geq S_{n-1}.$$

Um dies zu zeigen, wird der Algorithmus P zu einem Algorithmus P' modifiziert. Dies geschieht so, dass der neue Algorithmus P' den $(n-1)$-Kubus korrekt färbt. Dazu werden zunächst alle Schritte (C_i, a_i) mit $C_i = *C_i'$ entfernt. Anschließend wird bei den noch verbleibenden Unterkuben C_i die erste Komponente entfernt. Zum Schluss muss noch die Farbe a_i vertauscht werden, wenn $C_i = 1C_i'$ ist. Da der neue Algorithmus die Laufzeit $|P_0| + |P_1|$ hat, ist damit die noch übrige Behauptung $|P_0| + |P_1| \geq S_{n-1}$ gezeigt.

Zum vollständigen Abschluss des Beweises ist noch die Korrektheit der Modifikation von P zu P' zu zeigen: Als erstes werden bei der Modifikation alle Schritte (C_i, a_i) mit $C_i = *C_i'$ entfernt. Um dies zu erläutern, sei y ein Vektor aus dem $(n-1)$-Kubus und werde zum ersten Mal in Schritt i gefärbt. Es gilt dann, dass $C_i \neq *C_i'$, da sonst $0y$ und $1y$ dieselbe Farbe erhalten würden, was einen Widerspruch zur Korrektheit des Algorithmus darstellt. Daher ist ein weiterer Schritt mit einer Färbung von C_j' mit $C_j' = C_i'$ und $j > i$ nicht notwendig und muss daher entfernt werden, damit die Laufzeit des Algorithmus minimal wird.

Zum Schluss wird die Farbe a_i vertauscht, wenn $C_i = 1C_i'$ ist. Da C_i' eine 1 weniger enthält als $1C_i'$, ist diese Vertauschung notwendig, da die Farbe der Vektoren an die Anzahl der Einsen gekoppelt ist.

$\square$

5 Zusammenfassung

Zusammenfassend lässt sich Folgendes sagen: Die Frage, ob eine Menge färbbar ist, hängt von unterschiedlichen Bedingungen ab. Wie in Kapitel 3 gezeigt, kann der Durchschnitt der einzelnen Teilmengen ausschlaggebend sein. Aber auch die Größe und die Anzahl der einzelnen Teilmengen sind von Bedeutung.

Kapitel 4 weist auf die mögliche Komplexität von Färbungsalgorithmen im Allgemeinen hin und zeigt deren exponentielle Laufzeit auf.

Literatur

[1] Stasys Jukna. *Extremal Combinatorics*, chapter 6. Springer, 2001.