

Knotentheorie und ihre zentralen Forschungsfragen

Vorwissenschaftliche Arbeit verfasst von

Simon Grünbacher

8. Klasse

Betreuer: Mag. Johannes Teuschl

Februar 2020

Bundesrealgymnasium Linz,
4020 Linz, Hamerlingstraße 18

Abstract

Knoten sind in der realen Welt allgegenwärtig. Die Knotentheorie bemüht sich, grundlegende Eigenschaften von Knoten zu definieren und zu untersuchen. Dieses spannende Gebiet der Mathematik macht viele mathematische Disziplinen greifbar und zeigt deren Verbindungen auf.

In der Knotentheorie geht es darum, Knoten voneinander zu unterscheiden. Man muss also Knoten mathematisch definieren und festlegen, was sie voneinander unterscheidet. Weiters stellt sich die Frage, ob und wie Computerprogramme zur Unterscheidung von Knoten verwendet werden können.

In meiner vorwissenschaftlichen Arbeit gebe ich zunächst einen Überblick über wesentliche Konzepte und Forschungsfragen der Knotentheorie und spezialisiere mich dann auf das Thema Knotenfärbung. Ich stelle zwei neue Algorithmen zur Knotenfärbung vor, die auf den Prinzipien Backtracking und Divide-and-Conquer beruhen. Weiters analysiere ich die Performanz dieser Algorithmen auf Basis meiner Implementierung in der funktionalen Programmiersprache Haskell.

Inhaltsverzeichnis

Abstract	ii
1. Einleitung	1
2. Grundlagen	2
2.1. Knoten	2
2.2. Äquivalenz von Knoten	4
2.3. Knotendiagramme	5
2.4. Reidemeisterbewegungen	5
3. Forschungsfragen der Knotentheorie	7
3.1. Entknotungsproblem	7
3.2. Äquivalenzproblem	9
3.3. Auflistung von Knoten	10
3.4. Knoteninvarianten	11
4. Knotenfärbung	13
4.1. Dreifärbbarkeit	13
4.2. Fox'sche n -Färbbarkeit	14
4.3. Gruppen	16
4.4. Quandles	18
5. Algorithmen zur Knotenfärbung	23
5.1. Backtracking-Algorithmus	23
5.2. Divide-and-Conquer Algorithmus	25
6. Performanz und Korrektheit der Algorithmen	28
6.1. Forschungsfragen	28
6.2. Auswertung	29
6.3. Diskussion	31
7. Zusammenfassung und Ausblick	33

Inhaltsverzeichnis

A. Programmcode	35
Literatur	41

1. Einleitung

Wenn man zwei Seile hat, die jeweils in sich verknotet sind, stellt sich die Frage, ob man das eine in das andere umknoten kann, ohne eines der Enden irgendwo durchzuziehen. Dies ist die zentrale Frage der Knotentheorie.

Ursprünglich stammte das wissenschaftliche Interesse an Knoten aus einer Hypothese von Lord Kelvin, die besagt, dass die chemischen Eigenschaften der Elemente durch Verknotungen auf molekularer Ebene hervorgerufen werden.¹ Obwohl diese Theorie durch das moderne Atommodell widerlegt wurde, ist das theoretische Interesse an Knoten geblieben.

Im mathematischen Gebiet der Knotentheorie wird seit über 100 Jahren versucht, Knoten grundlegend zu formalisieren und deren Eigenschaften zu untersuchen. Um Knoten voneinander unterscheiden zu können, wird etwa versucht, Invarianten zu entdecken, die einen Knoten so identifizieren wie ein Fingerabdruck einen Menschen. Diese Fragen sind mathematisch höchst komplex. So gibt es Knoten, die zwar verschieden sind, allerdings durch bestimmte Invarianten nicht unterschieden werden können. Andererseits gibt es Invarianten, die zwar Knoten gut identifizieren können, allerdings sehr schwer zu berechnen sind.

In dieser vorwissenschaftlichen Arbeit stelle ich Theorien, Algorithmen und Implementierungen zur schnellen und zuverlässigen Berechnung von Invarianten zur Unterscheidung von Knoten vor. Zunächst führe ich in das Gebiet der Knotentheorie ein und diskutiere ihre wesentlichen Forschungsfragen. Danach stelle ich wesentliche Ansätze für Knoteninvarianten von Dreifärbbarkeit bis hin zu Quandles vor. Im praktischen Teil meiner Arbeit habe ich zwei Algorithmen zur Auffindung von Quandle-Färbungen entwickelt. Die weitere Arbeit umfasst eine Beschreibung der Funktionsweise dieser Algorithmen, ihre Implementierung in der funktionalen Sprache Haskell sowie eine Analyse ihrer Performanz.

¹vgl. Livingston, Charles (1995). *Knotentheorie für Einsteiger*. Vieweg Lehrbuch. Vieweg+Teubner Verlag. ISBN: 9783528066604, S. 1.

2. Grundlagen

In diesem Kapitel definieren wir zuerst, was ein Knoten überhaupt ist. Danach wird erläutert, was es bedeutet, wenn zwei Knoten gleich sind, also ineinander verformt werden können. Als nächstes sehen wir uns Knotendiagramme an, die zur Darstellung von Knoten verwendet werden. Das Kapitel endet mit dem Satz von Reidemeister, der der topologischen Definition von Knotenäquivalenz eine kombinatorische Formulierung verleiht, und uns so unserem Ziel der algorithmischen Unterscheidung von Knoten einen Schritt näher bringt.

2.1. Knoten

Im alltäglichen Sprachgebrauch ist ein Knoten ein Stück Seil oder Faden, das in sich selbst verschlungen ist. In unserer formalen Beschreibung ignorieren wir Dicke, Biogsamkeit und Reibung des Fadens, da es für einen Knoten prinzipiell keine Rolle spielt, ob er aus Drahtseilen oder Gitarrensaiten geknüpft ist. Um nicht bestimmte Materialparameter berücksichtigen zu müssen, gehen wir deshalb davon aus, dass der Faden unendlich dünn, beliebig flexibel und reibungsfrei ist. Unser Faden ähnelt also am ehesten einem Nähzwirn oder einer Drachenleine.

Unser Ziel ist es, Knoten voneinander unterscheidbar zu machen. Wenn wir einen Knoten als ein verschlungenes Stück Faden mit freien Enden definieren, dann können wir stets eines der Enden durch alle Schlaufen ziehen und so schließlich ein gerades Stück Faden erhalten. Es darf also keine freien Enden geben. Um das zu erreichen, könnten wir die Enden zusammenkleben. Dadurch erhalten wir eine geschlossene Schleife, die in sich verschlungen ist und bei der es kein Ende gibt, über das man die Schleife entwirren könnte. Solche Schleifen können wir dann besser voneinander unterscheiden.

Wir können eine solche Schleife mathematisch als eine geschlossene Kurve im dreidimensionalen Raum beschreiben. Zusätzlich fordern wir, dass sich diese Kurve nicht selbst schneidet. Insbesondere ist also auch eine leere Schlaufe ein Knoten, oft als *Unknoten* bezeichnet. Ein anderer Knoten ist in Abbildung 1 zu sehen. Dieser Knoten ist der einfachste „richtige“ Knoten. Bei den Seglern ist er als *halber Schlag* bekannt, in der Mathematik wird er als *Kleeblattschlinge* oder *Trefoil* bezeichnet.

2. Grundlagen

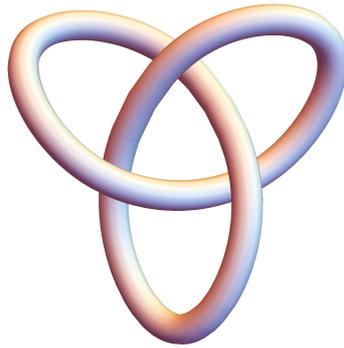


Abbildung 1.: Kleeblattschlinge.²

Die Kleeblattschlinge ist der Knoten, der entsteht, wenn man die Enden eines gewöhnlichen Knotens verklebt. Es scheint auf den ersten Blick so, als ließe sich dieser Knoten nicht zu einer einfachen Schleife entknoten, ohne irgendwo das Seil zu zerschneiden. Diese Vermutung stimmt auch, wie wir später noch beweisen werden.

So weit, so gut, doch da der Faden keine Ausdehnung hat, können wir ihn theoretisch auch unendlich oft verknoten. Das scheint sinnlos, ist per Definition aber noch möglich, weil eine Kurve in der Mathematik keine Ausdehnung hat. Solche unendlich verschlungenen Knoten werden als *wilde Knoten* bezeichnet und sind physikalisch nicht realisierbar.

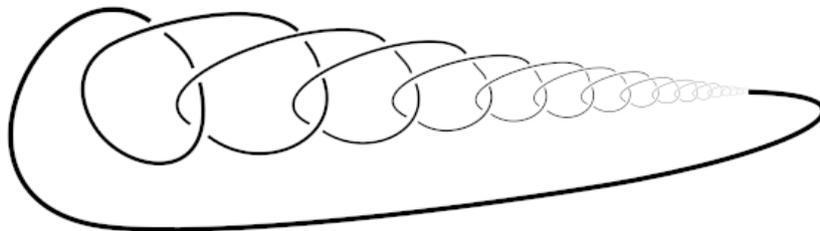


Abbildung 2.: Wilder Knoten.³

Um diese Möglichkeit auszuschließen, kann man zusätzlich noch verlangen, dass die Kurve *glatt* sein muss, was dazu führt, dass sie sich nicht unendlich stark krümmen kann, wodurch Fälle wie in Abbildung 2 nicht mehr möglich sind. Außerdem sollte die Kurve *regulär* sein, der Tangentialvektor also nie der Nullvektor wird. Daraus ergibt sich folgende Definition:

²Abbildung generiert mit Bar-Natan, Dror (2019). *KnotTheory*. http://katlas.org/wiki/The_Mathematica_Package_KnotTheory. (Besucht am 30. 10. 2019).

³contributors, Wikipedia (2015). *Wild knot* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Wild_knot&oldid=679495888. (Besucht am 30. 10. 2019).

Definition 2.1.1 (glatte Knoten⁴) *Ein glatter Knoten ist eine glatte, reguläre geschlossene Kurve im dreidimensionalen Raum, die sich nicht selbst schneidet.*

Es gibt verschiedene andere Definitionen von Knoten, die aber im Wesentlichen dasselbe beschreiben. Wir hätten einen Knoten etwa als geschlossenen, doppelpunktfreien endlichen Polygonzug definieren können. Diese Definition wird zum Beispiel oft beim Beweis des Satzes von Reidemeister verwendet. In der mathematischen Fachliteratur verwendet man aus technischen Gründen anstatt des dreidimensionalen Raums \mathbb{R}^3 oft \mathbb{S}^3 , also die dreidimensionale Oberfläche einer vierdimensionalen Kugel. Da diese Definitionen alle nur dazu dienen, eine ohnehin schon bestehende Vorstellung mathematisch zu fassen und der Unterschied nur in exotischen Fällen erkennbar ist, wollen wir uns damit hier nicht länger aufhalten.

2.2. Äquivalenz von Knoten

Eine der wichtigsten Fragen der Knotentheorie ist, ob zwei gegebene Knoten *äquivalent* sind. Damit soll gemeint sein, dass man einen Knoten in einen anderen umknoten kann, ohne das Seil zu zerschneiden. Zwei Dreiecke heißen *ähnlich*, wenn man die Ebene so verschieben, drehen und skalieren kann, dass das eine auf dem anderen aufliegt. Analog nennen wir im folgenden zwei Knoten *äquivalent*, wenn man den \mathbb{R}^3 so verformen kann, dass der eine Knoten auf dem anderen aufliegt. Wir beschreiben diese Verformung als eine Funktion $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. Wir möchten bei der Verformung keine Teile des Knotens reißen, deswegen sollte f *stetig* sein. Zusätzlich müssen wir noch ausschließen, dass die Funktion f den Raum spiegelt. Ähnlich wie bei Molekülen in der Chemie gibt es nämlich chirale Knoten, die von ihrem Spiegelbild verschieden sind, von denen es also zwei verschiedene Spiegelbilder gibt. Um solche chiralen Knotenpaare unterscheiden zu können, fordern wir, dass f *orientierungserhaltend* sein muss. Wenn der Knoten A äquivalent zum Knoten B ist, sollte B auch äquivalent zu A sein. Es sollte also zu f auch eine stetige Umkehrung g geben. Insgesamt gelangen wir zu folgender Definition:

Definition 2.2.1 (Knotenäquivalenz⁵) *Seien $A, B \subset \mathbb{R}^3$ zwei glatte Knoten. Dann heißen A und B äquivalent, wenn es Funktionen $f, g : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ gibt, sodass folgendes gilt:*

1. $f(g(x)) = g(f(x)) = x$.
2. $f(A) = B, g(B) = A$
3. f und g sind stetig.
4. f und g sind orientierungserhaltend.

⁴vgl. Kussin, Dirk (2005). *Knoten und Verschlingungen*. http://math-www.uni-paderborn.de/~dirk/Vorlesungen/Knoten/skript_12c.pdf. (Besucht am 07. 12. 2019), S. 4.

⁵vgl. ebd., S. 6.

2.3. Knotendiagramme

Oft ist es unbequem, für einen vorliegenden Knoten einen Funktionsterm zu finden, der die entsprechende Kurve im Raum beschreibt. Deshalb werden Knoten fast immer über Diagramme definiert. Dabei wird gewissermaßen der Schatten des Knotens in der Ebene dargestellt, wobei an Überkreuzungen vermerkt wird, welche Komponente oben ist.⁶ An einem Punkt dürfen sich maximal zwei Komponenten überkreuzen. Diese Projektion erlaubt es, Knoten schematisch darzustellen und zu definieren, ohne auf eine detaillierte Einbettung eingehen zu müssen. Jeder Knoten besitzt ein solches Diagramm. In Abbildung 3 ist ein Diagramm des Trefoil zu sehen.



Abbildung 3.: Knotendiagramm des Trefoil.⁷

2.4. Reidemeisterbewegungen

Diagramme reduzieren die Menge an unnötiger Information, die wir bei der Beschreibung eines Knotens benötigen, indem sie Details wie den Höhenunterschied zwischen unterkreuzendem und überkreuzendem Strang ignorieren. Jetzt, wo wir für Knoten eine kompaktere Darstellung gefunden haben, stellt sich die Frage, ob wir analog dazu auch für Knotenäquivalenz eine kompaktere Darstellung finden können. Kurt Reidemeister konnte diese Frage Anfang des 20. Jahrhunderts positiv beantworten: Er beschrieb die drei elementaren Bewegungen Ω_1 , Ω_2 , Ω_3 , die später als Reidemeisterbewegungen bekannt wurden (siehe Abbildung 4). Diese Bewegungen kann man nur auf einen Teil des Knotendiagramms anwenden, während man sonst nichts verändert. Durch Ω_1 ist es möglich, in ein gerades Seilstück eine Schlaufe zu drehen beziehungsweise eine Schlaufe zu entwirren. Ω_2 beschreibt, dass man zwei nebeneinander liegende Seilstücke an einer einzelnen Stelle überlappen kann, ohne etwas zu verändern. Mit Ω_3 kann man einen Strang, der auf der

⁶vgl. Kussin (2005), S. 6f.

⁷contributors, Wikipedia (2019). *Knot theory* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Knot_theory&oldid=925007816. (Besucht am 30. 10. 2019).

2. Grundlagen

einen Seite einer Kreuzung liegt, über diese Kreuzung auf die andere Seite heben. Reidemeister bewies folgendes:

Satz 2.4.1 (Kurt Reidemeister⁸, 1927) *Zwei Knotendiagramme stellen genau dann äquivalente Knoten dar, wenn sie sich durch eine endliche Folge von Reidemeisterbewegungen ineinander überführen lassen.*

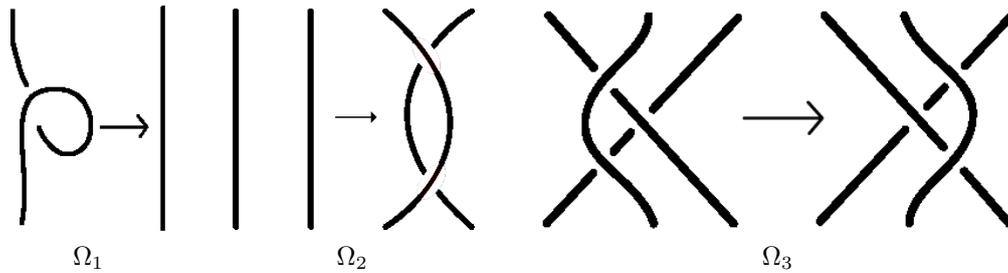


Abbildung 4.: Reidemeisterbewegungen.⁹

Auch wenn es so scheint, als wäre es dadurch einfach, Knotenäquivalenz festzustellen, so ist dem leider nicht so, wie wir im nächsten Kapitel sehen werden.

⁸vgl. Kussin (2005), S. 10.

⁹Eigenbearbeitung von contributors, Wikipedia (2017). *Tricolorability* — Wikipedia, The Free Encyclopedia. <https://en.wikipedia.org/w/index.php?title=Tricolorability&oldid=786521874>. (Besucht am 30.10.2019).

3. Forschungsfragen der Knotentheorie

Im Folgenden betrachten wir einige zentrale Fragen, die die Knotentheorie zu beantworten versucht. Die einfachste Frage zu einem Knoten ist, ob er entknotbar, also äquivalent zum Unknoten ist. Diese Frage ist als Entknotungsproblem bekannt und wird als erstes behandelt. Weiter geht es dann mit dem etwas allgemeineren Äquivalenzproblem, bei dem erkannt werden soll, ob zwei gegebene Knoten äquivalent sind. Ich beschreibe das Entknotungsproblem separat, da es dazu viele Forschungsergebnisse gibt, die sich nicht auf das Äquivalenzproblem verallgemeinern lassen. Das ursprüngliche Ziel der Knotentheorie war es, eine vollständige Liste aller Knoten zu erstellen. Im dritten Abschnitt wird der aktuelle Forschungsstand zu diesem Problem zusammengefasst. Eine Möglichkeit, das Äquivalenzproblem teilweise zu lösen, sind Knoteninvarianten. Diese werden im letzten Abschnitt behandelt. Im Anschluss daran werde ich mich auf Färbungsinvarianten spezialisieren.

3.1. Entknotungsproblem

Als Entknotungsproblem bezeichnet man die Frage, ob ein gegebener Knoten äquivalent zum Unknoten ist, sich also in eine leere Schleife verformen lässt. Falls ein Knoten entknotbar ist, dann ist die Entknotung kein zu langwieriges Unterfangen:

Satz 3.1.1 (Marc Lackenby¹⁰, 2015) *Wenn ein Knoten mit n Überkreuzungen entknotbar ist, dann ist das in $(236 * n)^{11}$ Reidemeisterbewegungen möglich.¹¹*

Falls ein Knoten nicht entknotbar ist, ist es ebenfalls stets möglich, dies in einem relativ kurzen Beweis zu begründen:

¹⁰vgl. Lackenby, Marc (2015). „A polynomial upper bound on Reidemeister moves“. In: *Annals of Mathematics* 182.2, S. 491–564. issn: 0003486X, Abstract.

¹¹Daraus folgt, dass das Entknotungsproblem in der Komplexitätsklasse *NP* liegt.

3. Forschungsfragen der Knotentheorie

Satz 3.1.2 (Marc Lackenby¹², 2016) *Es gibt Konstanten c, k , so dass folgendes gilt: Wenn ein Knoten mit n Überkreuzungen nicht entknotbar ist, dann lässt sich das in weniger als $c * n^k$ Schritten beweisen.¹³*

Es ist natürlich kaum praktikabel, alle Folgen von Reidemeisterbewegungen oder Beweisen dieser Länge durchzuprobieren, weswegen man typischerweise andere Methoden verwendet. Trotzdem zeigt dies, dass das Problem prinzipiell immer lösbar ist.

Eine spannende Frage, die man sich vielleicht zum Entknotungsproblem stellen könnte, ist folgende: Wenn ein Knoten entknotbar ist, kann man ihn dann so entknoten, dass die Zahl der Überkreuzungen im Diagramm die ursprüngliche Anzahl nie überschreitet, dass man ihn also nicht zwischendurch noch mehr verknoten muss?

Leider ist das nicht immer möglich, wie zum Beispiel der Culprit in Abbildung 5 zeigt. Der Knoten ist grundsätzlich entknotbar, muss aber erst komplizierter gemacht werden, um dann vereinfacht werden zu können.¹⁴ Es ist nämlich einerseits unmöglich, den Knoten mit den Reidemeisterbewegungen Ω_1 oder Ω_2 direkt zu vereinfachen, andererseits kann man auch nirgends die Reidemeisterbewegung Ω_3 anwenden.

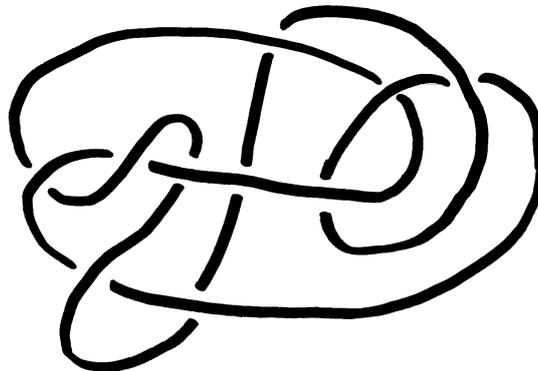


Abbildung 5.: Culprit.¹⁵

Immerhin kann man aber ein Diagramm des Unknoten mit n Kreuzungen stets so entknoten, dass die Anzahl der Kreuzungen nie $2 * (n + 1)^2$ überschreitet.¹⁶

¹²vgl. Lackenby, Marc (2016b). „The efficient certification of knottedness and Thurston norm“. In: *arXiv e-prints*, S. 1.

¹³Äquivalent dazu: Das Entknotungsproblem liegt in der Komplexitätsklasse *co-NP*.

¹⁴vgl. Kauffman, Louis H. und Lambropoulou, Sofia (2009). „Hard Unknots and Collapsing Tangles“. In: *Series on Knots and Everything – Introductory Lectures on Knot Theory*. Hrsg. von Kauffman, Louis H u. a. Bd. 46. World Scientific, S. 187–247. doi: 10.1142/9789814313001_0009, S. 3.

¹⁵Eigenarbeit auf Basis von ebd., Figure 2.

¹⁶vgl. Lackenby, Marc (2016a). „Elementary knot theory“. In: *arXiv e-prints*, S. 14.

3.2. Äquivalenzproblem

Wenn zwei Knotendiagramme äquivalente Knoten repräsentieren, kann man das auch beweisen, indem man Reidemeisterbewegungen durchprobiert, bis man eine passende Überführung gefunden hat. Das ist unter Umständen schwierig. Noch schwieriger ist es aber zu beweisen, dass zwei Diagramme *nicht* denselben Knoten darstellen. Denn wenn man keine entsprechende Folge von Reidemeisterbewegungen finden kann, ist das natürlich kein Beweis, dass es keine gibt. Was man also bräuchte, ist eine obere Grenze für die Anzahl der notwendigen Reidemeisterbewegungen für eine Umformung bei zwei gegebenen Diagrammen. Angenommen, man wüsste zum Beispiel für den Trefoil und den Achterknoten, dass eine Umformung, falls sie existiert, in weniger als 100 Reidemeisterbewegungen möglich wäre. Dann könnte man alle Folgen von weniger als 100 Reidemeisterbewegungen durchprobieren, um herauszufinden, dass keine davon vom Trefoil in den Achterknoten führt. Das würde beweisen, dass der Trefoil und der Achterknoten nicht äquivalent sind.

Tatsächlich kann man für zwei Diagramme D, D' stets eine solche obere Grenze finden. Genauer gilt:

Satz 3.2.1 (Marc Lackenby¹⁷, 2014) *Wenn zwei Knotendiagramme jeweils n und m Überkreuzungen haben, dann lassen sie sich durch eine Folge von weniger als $2^{2^{2^{\dots^{2^{n+m}}}}}$ Reidemeisterbewegungen ineinander überführen. Der Potenturm hat hierbei eine Höhe von $(_{10^{10^6}})^{(n+m)}$.*

Das Wachstum dieser Zahlen astronomisch zu nennen würde wohl selbst die Astronomie überfordern. Dementsprechend gibt es auch kaum Verfahren zur Überprüfung von Äquivalenz, die direkt auf Reidemeisterbewegungen beruhen, obwohl es natürlich sein könnte, dass es viel bessere Obergrenzen gibt als die von Marc Lackenby. Dennoch beweist diese Grenze, dass das Äquivalenzproblem grundsätzlich algorithmisch lösbar ist. Diese Tatsache wurde zuerst von Wolfgang Haken in den 1960er Jahren mithilfe eines Algorithmus bewiesen. Dieser Haken-Algorithmus ist allerdings sehr kompliziert und wird daher hier nicht behandelt. Wie viele Berechnungsschritte der Haken-Algorithmus maximal braucht, um Äquivalenz zu überprüfen, ist noch relativ unklar.¹⁸ Knotenäquivalenz wird dadurch zu einem spannenden algorithmischen Problem, weil einerseits unbekannt ist, wie lang man im schlimmsten Fall braucht, um es zu lösen, es aber andererseits praktische Verfahren gibt, die in den meisten Fällen relativ glaubwürdig erkennen, ob zwei Knoten äquivalent sind oder nicht. Implementierungen dieser Verfahren werden in Kapitel 5 näher behandelt.

¹⁷vgl. Lackenby (2016a), S. 16.

¹⁸vgl. ebd., S. 8.

3.3. Auflistung von Knoten

Nach Lord Kelvins Atomtheorie hängen chemische Eigenschaften von Elementen mit Verknotungen auf molekularer Ebene zusammen.¹⁹ Wenn diese Theorie auch falsch war, so war sie doch einer der Gründe, warum Knotentheorie überhaupt erst entstand, denn sie motivierte Peter Tait und Charles Little, eine Tabelle mit verschiedenen Knoten zu erstellen. Um 1900 hatten sie bereits eine Tabelle von allen Knoten mit weniger als 10 Überkreuzungen.²⁰

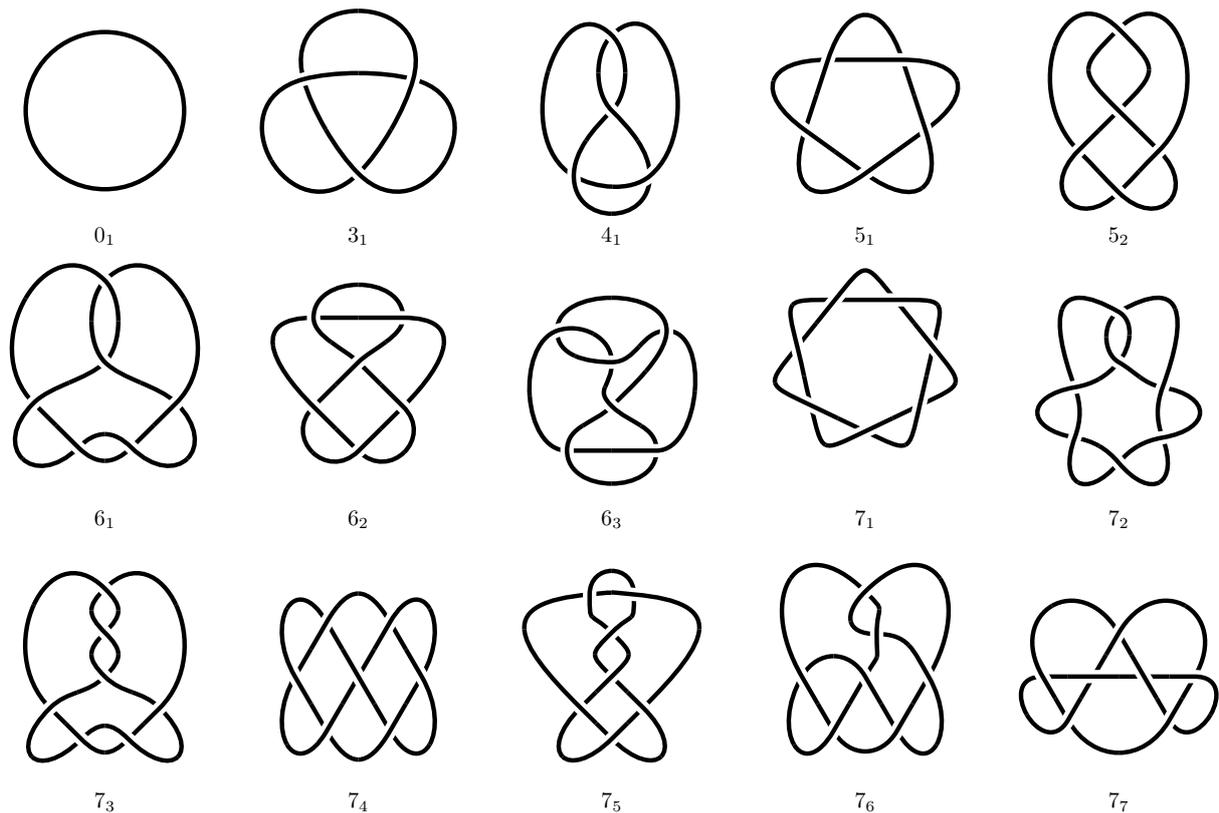


Abbildung 6.: Knotentabelle.²¹

Damals war der Formalismus, den wir im ersten Kapitel kennengelernt haben, noch weniger ausgereift. Trotzdem sind wir, was die prinzipielle Methode betrifft, kaum weiter als Tait und Little, denn bis heute ist der beste bekannte Ansatz zum Tabellieren von Knoten, alle Diagramme bis zu einer bestimmten Größe aufzuzeichnen und dann auf Äquivalenz zu filtern. Im Juli 2003 berechneten Stuart Rankin, Ortho Flint und John Schermann eine Tabelle mit allen 6, 217, 553, 258 Primknoten, die weniger als 22 Kreuzungen haben.²² Primknoten sind hierbei Knoten, die nicht

¹⁹vgl. Livingston (1995), S. 1.

²⁰vgl. ebd., S. 1.

²¹Abbildung generiert mit Bar-Natan (2019).

²²vgl. Menasco, William W. und (eds.), Morwen B. Thistlethwaite (2004). *Table of Contents for the Handbook of Knot Theory*, S. 2.

durch Zusammenhängen zweier anderer Knoten entstehen können. Wenn man eine Tabelle von Primknoten hat, kann man offensichtlich auch eine Tabelle von allen Knoten finden, indem man sie in allen Kombinationen zusammenhängt. Auf der Website *KnotInfo*²³ ist eine Tabelle von allen Knoten mit 12 oder weniger Kreuzungen verfügbar, jeweils zusammen mit einem Diagramm, den Knoten in verschiedenen Knotennotationen sowie Details zum Knoten.

3.4. Knoteninvarianten

Eine der vielversprechendsten Überlegungen zur Lösung des Äquivalenzproblems sind Knoteninvarianten. Eine Knoteninvariante wird typischerweise aus einem Diagramm berechnet und stellt gewissermaßen einen Fingerabdruck des Knotens dar, in dem Sinn dass jedes andere, äquivalente Diagramm zum selben Wert für die Invariante führt. Diese Invariante sollte also die Eigenschaft haben, dass sich ihr Wert unter Anwendung der Reidemeisterbewegungen nicht ändert. Wenn zwei Diagramme verschiedene Werte für die Invariante haben, ist das ein Beweis, dass sie nicht äquivalent sind, da jede Folge von Reidemeisterbewegungen, die vom einen Diagramm zum anderen führen würde, den Wert der Invariante erhalten würde. Umgekehrt folgt leider aus Gleichheit der Invarianten oft nicht die Äquivalenz der Knoten. Eine Invariante, die diese Eigenschaft erfüllt, wird *vollständig* genannt. Ein großer Teil der Arbeiten zur Knotentheorie beschäftigt sich mit der Suche nach möglichst vollständigen und berechenbaren Invarianten.

Eine der ersten Invarianten war das *Alexander-Polynom* (1928).²⁴ Anfang der 60er Jahre entwickelte Ralph Fox die *n-Färbbarkeit*, insbesondere die *Dreifärbbarkeit*, auf die wir im nächsten Kapitel eingehen werden.²⁵ Die Färbbarkeit von Fox war damals eigentlich ein Spezialfall der *Knotengruppe*, für die Wilhelm Wirtinger 1904 eine Formel entdeckte.²⁶ 1980 erfand David Joyce als Verallgemeinerung der Färbbarkeit den so genannten *Knotenquandle*. Dieser ist zwar bis auf umgekehrte Spiegelung eine vollständige Invariante, selber aber nur schwer auf Gleichheit überprüfbar.²⁷ 1985 entdeckte Vaughan Jones das *Jones-Polynom*²⁸ und etablierte damit auch eine Verbindung zwischen Knotentheorie und theoretischer Physik, für die ihm die Fields-Medaille

²³Cha, Jae Choon und Livingston, Charles (2019). *KnotInfo: Table of Knot Invariants*. <http://www.indiana.edu/~knotinfo>. (Besucht am 07. 12. 2019).

²⁴vgl. Livingston (1995), S. 3.

²⁵vgl. Przytyck, Józef H. (1998). „3-coloring and other elementary invariants of knots“. In: *Banach Center Publications* 42, S. 275–295. DOI: 10.4064/-42-1-275-295, S. 11.

²⁶vgl. Stillwell, John (2012). *Classical Topology and Combinatorial Group Theory*. Graduate Texts in Mathematics. Springer New York, S. 144.

²⁷vgl. Joyce, David (1982). „A classifying invariant of knots, the knot quandle“. In: *Journal of Pure and Applied Algebra* 23.1, S. 37–65. ISSN: 0022-4049, S. 61.

²⁸Jones, Vaughan F. R. (1985). „A polynomial invariant for knots via von Neumann algebras“. In: *Bulletin (New Series) of the American Mathematical Society* 12.1, S. 103–111.

3. Forschungsfragen der Knotentheorie

verliehen wurde. Wenig später entdeckten mehrere Autoren das *HOMFLY-Polynom*²⁹, welches eine Verallgemeinerung des Jones-Polynoms und des Alexander-Polynoms darstellt. In meiner Arbeit werde ich hauptsächlich auf Färbungsinvarianten eingehen und die Theorie der Polynomvarianten nicht weiter berücksichtigen, obwohl sich damit auch Bücher füllen lassen. Der interessierte Leser sei beispielsweise auf Kussin, Kauffman und Feichtner³⁰ verwiesen.

²⁹Freyd, P. u. a. (1985). „A new polynomial invariant of knots and links“. In: *Bulletin (New Series) of the American Mathematical Society* 12.2, S. 239–246.

³⁰Kussin (2005); Kauffman, Louis (1995). *Knoten – Diagramme, Zustandsmodelle, Polynomvarianten*. Heidelberg, Berlin, Oxford: Spektrum Akademischer Verlag. ISBN: 3860252321; Feichtner, Eva Maria (1994). „Polynomvarianten der Knotentheorie – Quellen ihrer Definition“. Magisterarb. FU Berlin.

4. Knotenfärbung

In diesem Kapitel beschreibe ich eine Hierarchie von verschiedenen Färbungsinvarianten. Ich werde mit der Dreifärbbarkeit beginnen und dann zeigen, wie sie sich zur Fox'schen n -Färbbarkeit verallgemeinern lässt. Als nächsten Schritt werde ich Gruppen-Färbung einführen und zeigen, wie die n -Färbbarkeit ein Spezialfall davon ist. In *Knotentheorie für Einsteiger*³¹ wird die Hierarchie von Färbungsregeln ähnlich eingeführt. Schließlich beschreibe ich Quandles, die im genannten Buch nicht vorkommen und eine weitere Verallgemeinerung der Gruppen-Färbung darstellen.

4.1. Dreifärbbarkeit

Wie der Name schon sagt, wird dabei überprüft, ob man einen Knoten in drei Farben (sagen wir rot, grün und blau) färben kann, sodass bestimmte Bedingungen erfüllt sind. Dabei bekommt jeder Bogen, also jede zusammenhängende Komponente im Diagramm, eine Farbe zugeordnet. An einer Kreuzung des Diagramms treffen dann immer drei farbige Bögen aufeinander. Damit eine Färbung eine Dreifärbung ist, müssen an jeder Kreuzung entweder drei verschiedene Farben oder eine einzige vorkommen, niemals aber zwei. Da man natürlich jeden Knoten einfarbig anmalen könnte, werden einfarbige Färbungen als *trivial* bezeichnet und nicht gezählt.

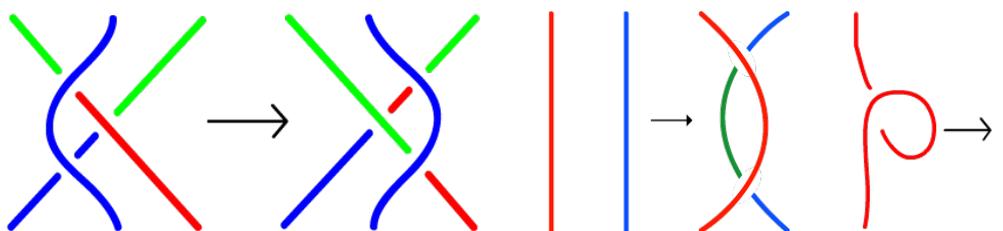


Abbildung 7.: Dreifärbungen unter Reidemeisterbewegungen.³²

³¹Livingston (1995).

³²contributors (2017).

4. Knotenfärbung

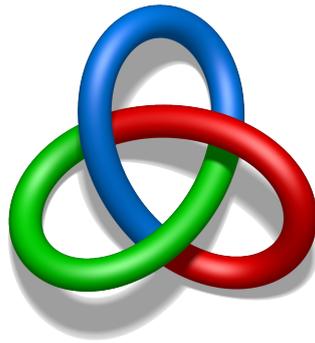


Abbildung 8.: Dreifärbung des Trefoil.³³

Wenn ein Knoten dreifärbbar ist und ein anderer nicht, lässt sich daraus schließen, dass die beiden Knoten verschieden sind. Es lässt sich nämlich relativ leicht zeigen, dass bei Anwendung von Reidemeisterbewegungen die Anzahl der Dreifärbungen erhalten bleibt. Eine Skizze des Beweises ist in Abbildung 7 gegeben. Darin sieht man, dass eine bestimmte Färbung der an der Bewegung beteiligten Bögen unter Anwendung der Reidemeisterbewegungen erhalten werden kann. Man kann diese Tatsache auch für alle anderen möglichen Färbungen der beteiligten Bögen überprüfen.

Der Trefoil ist dreifärbbar, wie Abbildung 8 zeigt. Der Unknoten und der Achterknoten hingegen sind nicht dreifärbbar, also verschieden vom Trefoil.

4.2. Fox'sche n -Färbbarkeit

Der Achterknoten ist nicht dreifärbbar. Wir wissen daher nicht sicher, ob er entknotbar ist oder nicht. Um das herauszufinden versuchen wir, die Idee der Dreifärbbarkeit zu verallgemeinern.

Wir brauchen dazu entweder mehr Farben oder andere Bedingungen, die wir an den Kreuzungen stellen. Es stellt sich allerdings heraus, dass es bei drei Farben nur eine Menge von Bedingungen gibt, die in einer sinnvollen Knoteninvariante resultieren. Deshalb werden wir es mit mehr Farben versuchen. Doch welche Bedingungen müssen wir an den Kreuzungen einfordern, um eine Knoteninvariante zu erhalten? Wenn wir bei der Dreifärbbarkeit den Farben in beliebiger Reihenfolge die Zahlen 0, 1 und 2 zuordnen, können wir bei den Kreuzungen folgendes Muster feststellen: Die Summe der Farbindizes der unterkreuzenden Bögen ist modulo 3 gleich dem doppelten Farbindex des überkreuzenden Bogen. Wenn die drei Indizes gleich sind, ist das natürlich klar, weil $x + x \equiv 2x \pmod{3}$ immer gilt. Wenn die Indizes alle verschieden sind, erhalten wir beispielsweise die folgenden Gleichungen:

³³Eigenbearbeitung von contributors (2017).

4. Knotenfärbung

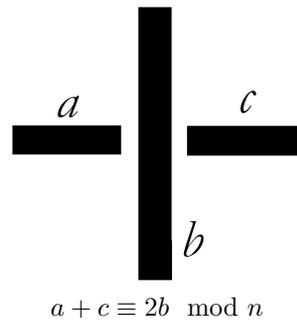


Abbildung 9.: Bedingung an den Kreuzungen.³⁴

1. $1 + 2 \equiv 2 * 3 \pmod{3}$
2. $1 + 3 \equiv 2 * 2 \pmod{3}$
3. $2 + 3 \equiv 2 * 1 \pmod{3}$

Das Muster hält also auch für sämtliche Kombinationen an verschiedenen Farben.

Dieses Muster lässt sich auch auf andere Zahlen als 3 ausweiten. Tatsächlich können wir für jede natürliche Zahl n eine n -Färbung definieren, deren Farbindizes die Gleichung $a + c \equiv 2b \pmod{n}$ an jeder Kreuzung erfüllen (siehe Abbildung 9). Das Ergebnis ist stets eine neue Knoteninvariante, nämlich genau die Fox'sche n -Färbbarkeit. Wie in Abbildung 10 sichtbar, ist es möglich, für den Achterknoten eine nichttriviale 5-Färbung zu finden, was beweist, dass der Achterknoten nicht entknotbar ist.³⁵ Fox- n -Färbungen stellen allerdings keine vollständige Invariante dar. Es

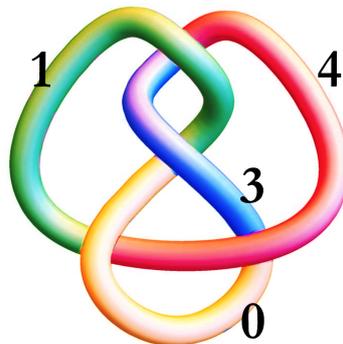


Abbildung 10.: Fünffärbung des Achterknotens.³⁶

gibt also Knoten, die nicht äquivalent sind, allerdings für jede Zahl n gleich viele n -Färbungen besitzen. Insbesondere gibt es auch Knoten, die nicht äquivalent zum Unknoten sind, allerdings keine nichttrivialen n -Färbungen besitzen.

³⁴Eigenarbeit, vgl. Livingston (1995), S. 37.

³⁵vgl. Fish, Andrew, Lisitsa, Alexei und Stanovský, David (2015). „A Combinatorial Approach to Knot Recognition“. In: *Embracing Global Computing in Emerging Economies*. Hrsg. von Horne, Ross. Cham: Springer International Publishing, S. 64–78. ISBN: 978-3-319-25043-4, S. 4.

³⁶Eigenarbeit auf Basis von <http://katlas.org/> und ebd., Figure 1.

4.3. Gruppen

Eine *Gruppe* ist in der Mathematik eine Menge von Bewegungen, die man auf ein Objekt anwenden kann. Diese Bewegungen sollte man nacheinander ausführen und umkehren können. Beispielsweise bildet die Menge an Bewegungen eines Rubik's Cube eine Gruppe, ebenso wie die Möglichkeiten, ein Dreieck anders hinzulegen, sodass es danach denselben Schatten hat. Da das Objekt, das bewegt wird, selbst nicht von Interesse ist und Gruppen auch in vielen anderen Zusammenhängen zum Einsatz kommen, kommt es in der formalen Definition nicht vor und es wird stattdessen nur die Menge der Bewegungen G , die Operation des Umkehrens $^{-1}$ und die Operation des Nacheinander-Ausführens $*$ berücksichtigt.

Definition 4.3.1 (Gruppe³⁷) *Eine Gruppe ist eine Menge G , zusammen mit einer binären Operation $*$, und einer unären Operation $^{-1}$, so dass die folgenden Bedingungen erfüllt sind:*

1. $(a * b) * c = a * (b * c)$ für alle $a, b, c \in G$
2. Es gibt eine *Einheit* $e \in G$, sodass $e * a = a$ für alle $a \in G$ gilt. Statt e wird oft 1 geschrieben.
3. Für jedes $a \in G$ gilt $a * a^{-1} = e$. a^{-1} wird auch als *inverses Element von a* bezeichnet.

Ein Beispiel für eine Gruppe sind wie erwähnt die Bewegungen, die ein Dreieck so bewegen, dass es danach denselben Schatten hat wie davor. Dabei können wir ein Dreieck um 120, 240 oder 360 Grad drehen und um jede der Ecken spiegeln. Es reicht aber auch aus, eine Ecke zum Spiegeln festzulegen und das Dreieck vor und nach jeder Spiegelung zurecht zu drehen. Diese fixierte Spiegelung bezeichnen wir mit b_1 . Wir könnten das Dreieck auch um 120 oder 240 Grad im Uhrzeigersinn drehen und dann spiegeln, diese Bewegungen bezeichnen wir mit b_2 und b_3 . Für diese drei Elemente gelten die folgenden Gleichungen:

1. $b_1 * b_2 * b_1^{-1} = b_2 * b_1 * b_2^{-1} = b_3$
2. $b_1 * b_3 * b_1^{-1} = b_3 * b_1 * b_3^{-1} = b_2$
3. $b_3 * b_2 * b_3^{-1} = b_2 * b_3 * b_2^{-1} = b_1$

Wir können Dreifärbbarkeit auch über die Elemente b_1, b_2, b_3 definieren. Es bekommt also jeder der Bögen eines dieser drei Gruppenelemente zugeordnet, wobei an jeder Kreuzung, an der der Bogen a von links aus unter den Bogen b führt und rechts als Bogen c wieder heraus kommt, gelten muss: $b * a * b^{-1} = c$ (siehe Abbildung 11). Um von links und rechts sprechen zu können, brauchen die Bögen eine Fahrtrichtung. Diese kann beliebig gewählt werden und ist für die endgültige Konstruktion nicht wichtig. Dieses Muster ist nicht exklusiv für Dreiecke: Es gibt zu jedem Polygon eine Gruppe von Drehungen, die so genannte *Diedergruppe*. Aus jeder dieser

³⁷vgl. Fischer, Gerd (2005). *Lineare Algebra*. Springer Spektrum. ISBN: 978-3-658-03944-8, S. 43.

4. Knotenfärbung

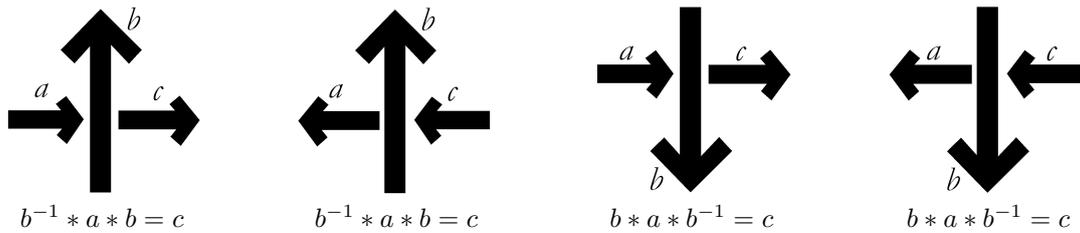


Abbildung 11.: Bedingungen an den Kreuzungen.³⁸

Diedergruppen ergibt sich eine neue Färbungsinvariante. Allerdings ist diese gar nicht so neu, sie entspricht nämlich genau der n -Färbbarkeit von Fox!

Polygone sind nicht die einzige Möglichkeit, Gruppen zu definieren. Ein anderes Beispiel für eine Gruppe wäre die der Bewegungen des bereits erwähnten Rubik's Cube, es gibt allerdings noch zahllose andere Gruppen. Jede davon gibt uns eine andere Färbungsregel, die wir zur Unterscheidung von Knoten verwenden können. Tatsächlich gibt es für jeden nicht-entknotbaren Knoten K eine Gruppe G , sodass K eine nichttriviale G -Färbung besitzt.³⁹ Dennoch gibt es Knoten, die über Gruppen nicht voneinander unterschieden werden können, zum Beispiel den Square Knot und den Granny Knot.⁴⁰ Wir könnten nämlich die Färbung als ein Gleichungssystem

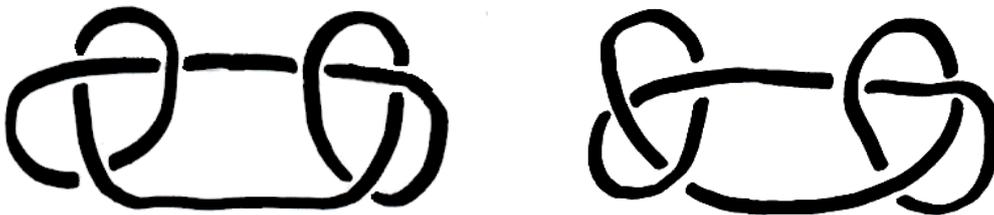


Abbildung 12.: Square Knot und Granny Knot.⁴¹

verstehen, wobei jeder Bogen eine Variable darstellt und jede Kreuzung eine Gleichung der Form $b * a * b^{-1} = c$. Man kann beweisen, dass das Gleichungssystem des Square Knot in das des Granny Knot umgeformt werden kann.⁴² Daraus folgt, dass sie in jeder Gruppe gleich viele Färbungen besitzen. Dennoch sind die beiden Knoten nicht äquivalent, wie wir im nächsten Kapitel sehen werden.

³⁸Eigenarbeit auf Basis von Livingston (1995), S.81f.

³⁹vgl. Kuperberg, Greg (2014). „Knottedness is in NP, modulo GRH“. In: *Advances in Mathematics* 256, S. 493–506. ISSN: 0001-8708, S. 6.

⁴⁰vgl. Feichtner (1994), S. 26.

⁴¹Eigenarbeit auf Basis von ebd., S. 25.

⁴²vgl. ebd., S. 26.

Obwohl es Paare von unterschiedlichen Knoten gibt, deren Gruppen-Gleichungssysteme ineinander umgeformt werden können, ist das immerhin für unterschiedliche Primknoten, also Knoten, die nicht durch Zusammenhängen kleinerer Knoten entstehen, nie der Fall:

Satz 4.3.1 (Wilbur Whitten⁴³, 1987) *Wenn A und B zwei Primknoten sind, deren Gruppen-Gleichungssysteme ineinander umgeformt werden können, dann sind A und B oder A und das Spiegelbild von B äquivalent.*⁴⁴

Offenbar sind der Square Knot und der Granny Knot keine Primknoten, weil sie durch Verkettung zweier Trefoils entstehen. Dabei sind diese beim Granny Knot gleich orientiert und beim Square Knot unterschiedlich. Daher kann man den Satz von Whitten hier nicht anwenden.

4.4. Quandles

Der *Quandle* ist die allgemeinste Form von Färbungsregeln. Der Name *Quandle* geht auf David Joyce⁴⁵ zurück, der in seiner Doktorarbeit bewies, dass das *Quandle-Gleichungssystem* eine vollständige Knoteninvariante ist.⁴⁶ Auch vor dieser Doktorarbeit wurden schon ähnliche Strukturen erforscht. Beispiele sind der *Kei*⁴⁷, der *Kristall*⁴⁸ oder der *Rack*⁴⁹. Der Quandle wurde im Zusammenhang mit Knoten entwickelt, um Gruppen-Färbung zu verallgemeinern. Damit ein Knoten eine G -Färbung für eine Gruppe G besitzt, müssen bekanntlich Gleichungen der Form $y^{-1} * x * y$ erfüllt sein (siehe Abbildung 11). Diese Operation wird oft auch als Konjugation bezeichnet. Wir könnten eigene Symbole für Konjugation definieren, zum Beispiel $x \triangleright y := y^{-1} * x * y$ und $x \triangleleft y := y * x * y^{-1}$.⁵⁰ Da wir nur Konjugation brauchen, um die Gleichungen im Knoten anzuschreiben, können wir die Gruppenstruktur von G einfach vergessen und statt $*$, 1 und $^{-1}$ einfach \triangleright und \triangleleft zu fundamentalen Größen erheben. In einer Gruppe müssen $*$, 1 und $^{-1}$ bestimmte Bedingungen erfüllen (siehe Definition 4.3.1). Es müssen also auch \triangleright und \triangleleft bestimmte Bedingungen erfüllen, die sich in folgender Definition zusammenfassen lassen:

⁴³vgl. Whitten, Wilbur (1987). „Knot Complements and Groups“. In: *Topology* 26.1, S. 41–11, S. 41.

⁴⁴In der zitierten Arbeit wird ein Satz bewiesen, der vom wenig später bewiesenen Gordon-Luecke Theorem (Gordon, C. McA. und Luecke, J. (1989). „Knots are determined by their complements“. In: *Bulletin (New Series) of the American Mathematical Society* 20.1. [21], S. 83–87, S. 1) abhängt.

⁴⁵vgl. Joyce (1982).

⁴⁶vgl. ebd., S. 65.

⁴⁷vgl. ebd., S. 38.

⁴⁸vgl. Kauffman (1995), S. 157.

⁴⁹vgl. Fenn, Roger und Rourke, Colin (1992). „Racks and links in codimension two“. In: *Journal of Knot Theory and Its Ramifications* 1.4, S. 343–406, S. 343.

⁵⁰vgl. Joyce (1982), S.37.

4. Knotenfärbung

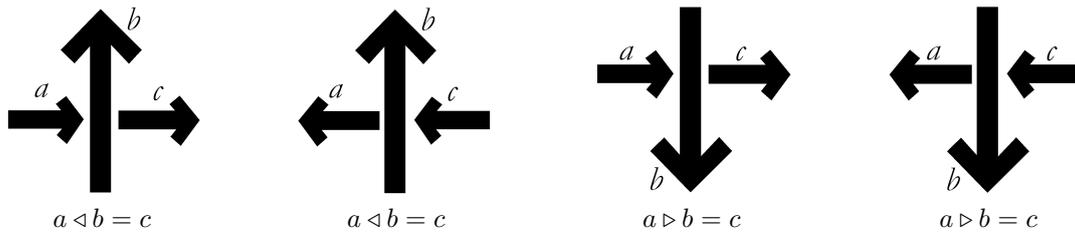


Abbildung 13.: Definition der Quandle-Operationen im Knoten.⁵²

Definition 4.4.1 (Quandle⁵¹) Ein Quandle ist eine Menge Q zusammen mit einer Operation \triangleright und deren inverser Operation \triangleleft . Diese Operationen erfüllen dann die folgenden Gleichungen für alle $x, y, z \in Q$.

1. $x \triangleright x = x$
2. $(x \triangleright y) \triangleleft y = x = (x \triangleleft y) \triangleright y$
3. $(x \triangleright y) \triangleright z = (x \triangleright z) \triangleright (y \triangleright z)$

Der Quandle heißt endlich, wenn die Menge Q endlich ist.

Offensichtlich kann jede Gruppe G zu einem Quandle gemacht werden, wenn man $x \triangleright y := y^{-1} * x * y$ und $x \triangleleft y := y * x * y^{-1}$ definiert. Das sollte auch so sein, denn das war ja Ausgangspunkt unserer Definition von Quandles. Es gibt allerdings auch Quandles, die nicht auf diese Weise aus Gruppen entstehen.

Die Knotenfärbung über Gruppen lässt sich verallgemeinern zu einer Knotenfärbung über Quandles. Dabei werden Bögen des Knoten mit Elementen von Q eingefärbt, und an jeder Kreuzung muss eine bestimmte Gleichung erfüllt sein. Wenn der Bogen a rechts von b ist und der Bogen c links von b ist muss $a \triangleright b = c$ gelten. Ist a links von b und c rechts von b , muss $a \triangleleft b = c$ gelten (siehe Abbildung 13). Wie im vorherigen Kapitel brauchen die Bögen eine Fahrtrichtung, damit man rechts und links unterscheiden kann. In diesem Fall spielt diese Fahrtrichtung allerdings eine Rolle für die Quandle-Färbbarkeit.⁵³

Die Quandleaxiome scheinen auf den ersten Blick sehr beliebig zu sein, doch wenn wir die Gleichungen in der beschriebenen Weise aufzeichnen, wird erkennbar, dass sie genau die Reidemeisterbewegungen codieren (siehe Abbildung 14). Ähnlich wie bei den Gruppen kann man auch

⁵¹vgl. Joyce (1982), S. 37.

⁵²Eigenarbeit, vgl. ebd., S. 57.

⁵³vgl. Clark, W. E. u. a. (2014). „Quandle colorings of knots and applications“. In: *Journal of Knot Theory and Its Ramifications* 23 (6). doi: 10.1142/S0218216514500357, Lemma 3.2.

⁵⁴Eigenbearbeitung von contributors (2017).

4. Knotenfärbung

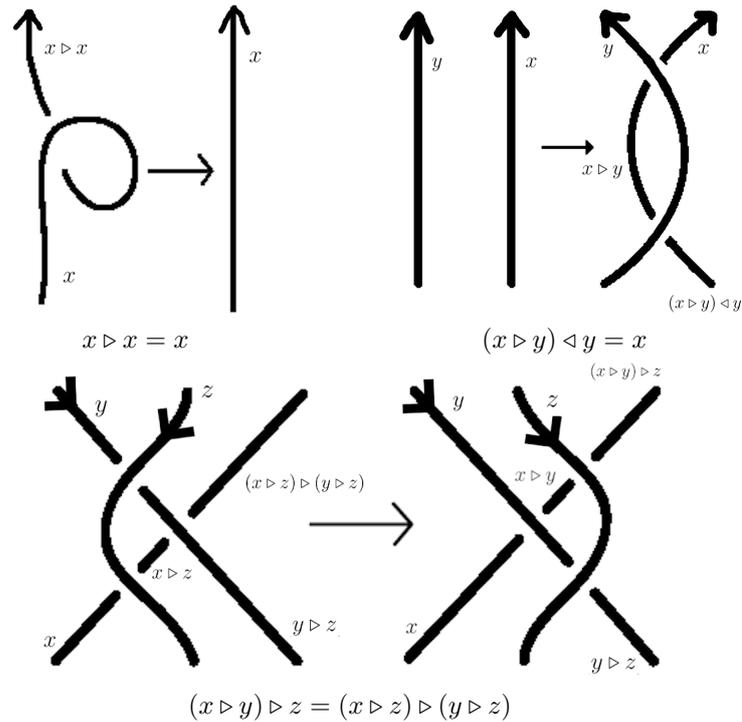


Abbildung 14.: Quandleaxiome im Knoten für die drei Reidemeisterbewegungen.⁵⁴

für Quandles jedem Knoten ein Gleichungssystem zuzuordnen. Dabei wird wieder jedem Bogen eine Variable und jeder Kreuzung eine Gleichung zugewiesen. Für den Trefoil wäre das zum Beispiel $x \triangleright y = z$, $z \triangleright x = y$, $y \triangleright z = x$. Diese Gleichungssysteme kann man auf zwei Arten umformen:⁵⁵

1. Indem man eine Gleichung, die aus den anderen Gleichungen folgt, hinzufügt oder entfernt.
2. Indem man eine neue Variable hinzufügt oder entfernt, gemeinsam mit einem Term, der diese definiert.

Zum Beispiel könnte man das Gleichungssystem des Trefoil umformen, indem man die triviale Gleichung $x \triangleright x = x$ hinzufügt, oder indem man eine Variable t gemeinsam mit einer definierenden Gleichung $t = x \triangleright (y \triangleleft z)$ hinzufügt. David Joyce bewies 1982 folgendes:

Satz 4.4.1 (David Joyce⁵⁶, 1982) *Wenn die Gleichungssysteme zweier Knoten A und B ineinander umgeformt werden können, dann sind A und B oder A und das umgekehrte Spiegelbild von B orientiert äquivalent. Wenn zwei Knoten bis auf umgekehrte Spiegelung orientiert äquivalent sind, können die entsprechenden Gleichungssysteme auch ineinander umgeformt werden. Das Quandle-Gleichungssystem eines orientierten Knotens ist also bis auf umgekehrte Spiegelung eine vollständige Knoteninvariante.*

⁵⁵vgl. Fenn und Rourke (1992), S. 365.

⁵⁶vgl. Fish, Lisitsa und Stanovský (2015), S. 3.

4. Knotenfärbung

Mit einer umgekehrten Spiegelung ist hierbei gemeint, dass man die Fahrtrichtung des Knotens umkehrt und ihn dann spiegelt. Mit orientiert äquivalent ist gemeint, dass die in der Definition von Äquivalenz verwendete Funktion die Fahrtrichtung des Knotens erhält. Um zu beweisen, dass zwei Knoten verschieden sind, müsste man zeigen, dass sich die Gleichungssysteme nicht ineinander umformen lassen. Da dies schwer beweisbar ist, versucht man über Färbungen, die Gleichungssysteme in bestimmten endlichen Quandles zu lösen. Wenn sie dabei unterschiedlich viele Lösungen haben, sind die Gleichungssysteme und damit die Knoten verschieden. Die Frage, ob es für zwei verschiedene Knoten immer einen endlichen Quandle gibt, der diese dadurch unterscheidet, ist leider noch offen.⁵⁷ Es war also bisher für alle untersuchten Paare von Knoten möglich, einen Quandle zu finden, der sie unterscheidet. Andererseits ist es denkbar, dass es Knoten gibt, die zwar unterschiedlich sind, von der Quandle-Färbung allerdings nicht unterschieden werden können.

Dreifärbbarkeit lässt sich als Quandle realisieren, wobei $Q = \{r, g, b\}$ und $r \triangleright r = r$, $r \triangleright g = b$, $r \triangleright b = g$, $g \triangleright r = b$, etc. Mit etwas Mühe kann man überprüfen, dass alle Bedingungen für einen endlichen Quandle erfüllt sind. Eine Dreifärbung für einen Knoten zu finden bedeutet also, im rgb-Quandle das Quandle-Gleichungssystem des Knoten zu lösen.

Die Fox'sche n -Färbbarkeit ist als Quandle realisierbar, wenn man $Q = \{1, \dots, n\}$ setzt und $a \triangleright b = a \triangleleft b = 2b - a \pmod n$. Um ein Quandle-Gleichungssystem in der n -Färbbarkeit zu lösen, muss man also im Wesentlichen ein Lineargleichungssystem in modularer Arithmetik lösen. Diese Gleichungssysteme sind über den chinesischen Restsatz relativ leicht lösbar.⁵⁸ Es ist aber nicht immer möglich, ein Quandle-Gleichungssystem in ein Lineargleichungssystem umzuwandeln.

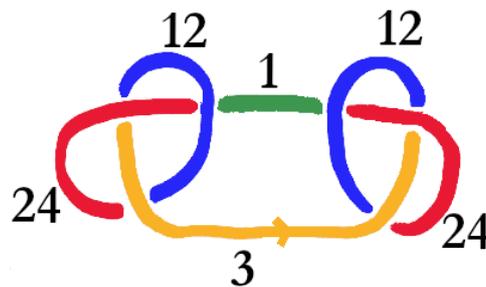


Abbildung 15.: Färbung des Square Knot.

Im vorigen Kapitel sind wir daran gescheitert, den Square Knot und den Granny Knot voneinander zu unterscheiden. Quandle-Färbung gibt uns eine Möglichkeit, das zu tun. Es gibt nämlich einen Quandle, in dem man den Square Knot färben kann, denn Granny Knot allerdings nicht (siehe Abbildung 15). Die Wertetabelle der Operation \triangleright in diesem Quandle ist in Tabelle 4.1 gegeben.

⁵⁷vgl. Fish, Lisitsa und Stanovský (2015), S. 13.

⁵⁸vgl. ebd., S. 5.

4. Knotenfärbung

Tabelle 4.1.: Wertetabelle des Quandles.⁵⁹

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1	1	1	1	1	3	3	3	3	24	24	24	24	18	18	18	18	12	12	12	12	15	15	15	15	
2	2	2	2	2	4	4	4	4	23	23	23	23	17	17	17	17	11	11	11	11	16	16	16	16	
3	3	3	3	3	2	2	2	2	21	21	21	21	20	20	20	20	9	9	9	9	14	14	14	14	
4	4	4	4	4	1	1	1	1	22	22	22	22	19	19	19	19	10	10	10	10	13	13	13	13	
5	7	7	7	7	5	5	5	5	17	17	17	17	24	24	24	24	14	14	14	14	10	10	10	10	
6	8	8	8	8	6	6	6	6	18	18	18	18	23	23	23	23	13	13	13	13	9	9	9	9	
7	6	6	6	6	7	7	7	7	19	19	19	19	21	21	21	21	16	16	16	16	12	12	12	12	
8	5	5	5	5	8	8	8	8	20	20	20	20	22	22	22	22	15	15	15	15	11	11	11	11	
9	20	20	20	20	23	23	23	23	9	9	9	9	11	11	11	11	8	8	8	8	2	2	2	2	
10	19	19	19	19	24	24	24	24	10	10	10	10	12	12	12	12	7	7	7	7	1	1	1	1	
11	17	17	17	17	22	22	22	22	11	11	11	11	10	10	10	10	5	5	5	5	4	4	4	4	
12	18	18	18	18	21	21	21	21	12	12	12	12	9	9	9	9	6	6	6	6	3	3	3	3	
13	22	22	22	22	18	18	18	18	15	15	15	15	13	13	13	13	1	1	1	1	8	8	8	8	
14	21	21	21	21	17	17	17	17	16	16	16	16	14	14	14	14	2	2	2	2	7	7	7	7	
15	24	24	24	24	20	20	20	20	14	14	14	14	15	15	15	15	3	3	3	3	5	5	5	5	
16	23	23	23	23	19	19	19	19	13	13	13	13	16	16	16	16	4	4	4	4	6	6	6	6	
17	16	16	16	16	10	10	10	10	4	4	4	4	7	7	7	7	7	17	17	17	17	19	19	18	19
18	15	15	15	15	9	9	9	9	3	3	3	3	8	8	8	8	18	18	18	18	20	20	20	20	
19	13	13	13	13	12	12	12	12	1	1	1	1	6	6	6	6	19	19	19	19	18	18	18	18	
20	14	14	14	14	11	11	11	11	2	2	2	2	5	5	5	5	20	20	20	20	17	17	17	17	
21	9	9	9	9	16	16	16	16	6	6	6	6	2	2	2	2	23	23	23	23	21	21	21	21	
22	10	10	10	10	15	15	15	15	5	5	5	5	1	1	1	1	24	24	24	24	22	22	22	22	
23	11	11	11	11	13	13	13	13	8	8	8	8	4	4	4	4	22	22	22	22	23	23	23	23	
24	12	12	12	12	14	14	14	14	7	7	7	7	3	3	3	3	21	21	21	21	24	24	24	24	

⁵⁹Stanovský, David (2019). *Calculating with Quandles*. <http://www.karlin.mff.cuni.cz/~stanovsk/quandles/>. (Besucht am 30. 10. 2019), 'multiplication tables of all connected quandles of order less than 36'.

5. Algorithmen zur Knotenfärbung

Nun stellt sich natürlich die Frage, wie man bestimmt, ob für einen gegebenen Knoten und Quandle eine Färbung existiert. Die direkte Möglichkeit wäre, alle Färbungen aufzulisten und zu überprüfen, ob eine davon alle Gleichungen löst. Für einen Quandle mit m Farben und einen Knoten mit n Bögen gibt es allerdings m^n verschiedene Färbungen, die man alle überprüfen müsste. Bei einem Knoten mit 85 Bögen und einem Quandle mit 10 Farben gäbe es also schon mehr Färbungen zu überprüfen als Teilchen im beobachtbaren Universum! Man braucht also offensichtlich ein effizienteres Verfahren.

Zuerst einmal gilt es, das Problem zu konkretisieren. Es gibt offensichtlich einerseits Bögen, die bestimmte Farben haben können. Wir wollen eine Färbung finden, die an jeder Kreuzung die Bedingung $a \triangleright b = c$ erfüllen sollte. Wir gehen davon aus, dass wir die Werte von $a \triangleright b$ in einer Tabelle gegeben haben.

Es gibt noch eine zweite Möglichkeit, das Problem zu beschreiben: Anstatt zu sagen, dass wir Bögen färben und an den Kreuzungen Bedingungen erfüllen müssen, können wir auch sagen, dass wir Kreuzungen färben und an den Bögen Bedingungen erfüllen müssen. Wir können eine Kreuzung einfärben, indem wir ihr ein Tripel (a, b, c) mit $a \triangleright b = c$ zuordnen. Da ein Bogen nur eine Farbe hat, muss er durch alle Kreuzungen gleich eingefärbt werden. Dadurch gibt es eine Bedingung, die wir an den Bögen erfüllen müssen.

Ich habe für beide Beschreibungen des Problems ein Backtracking-Verfahren entwickelt und implementiert. Experimentell gesehen sind beide Algorithmen vergleichbar schnell und ähnlich komplex in ihrer Implementierung. Im folgenden Abschnitt werde ich mich daher auf die zweite Beschreibung beschränken.

5.1. Backtracking-Algorithmus

Eine gängige Klasse von Verfahren zur Lösung von Problemen mit exponentiell großen Suchräumen sind Backtracking-Algorithmen. Die Idee ist simpel: Angenommen, wir suchen nach einer Dreifärbung für den Achterknoten. Wenn wir die erste Kreuzung einfärben, haben wir

5. Algorithmen zur Knotenfärbung

9 Möglichkeiten. Wenn wir als zweites irgendeine Kreuzung auswählen würden, hätten wir jeweils wieder 9 Möglichkeiten, also insgesamt 81. Wenn wir allerdings eine direkt benachbarte Kreuzung wählen, wie in Abbildung 16 dargestellt, haben wir nur maximal eine Möglichkeit, sie zu färben. In 6 Fällen haben wir gar keine Möglichkeit. Am Ende bleiben uns 3 vollständige Färbungen des Achterknoten übrig, die alle einfarbig sind. Damit haben wir gezeigt, dass der Achterknoten nicht dreifärbbar ist.

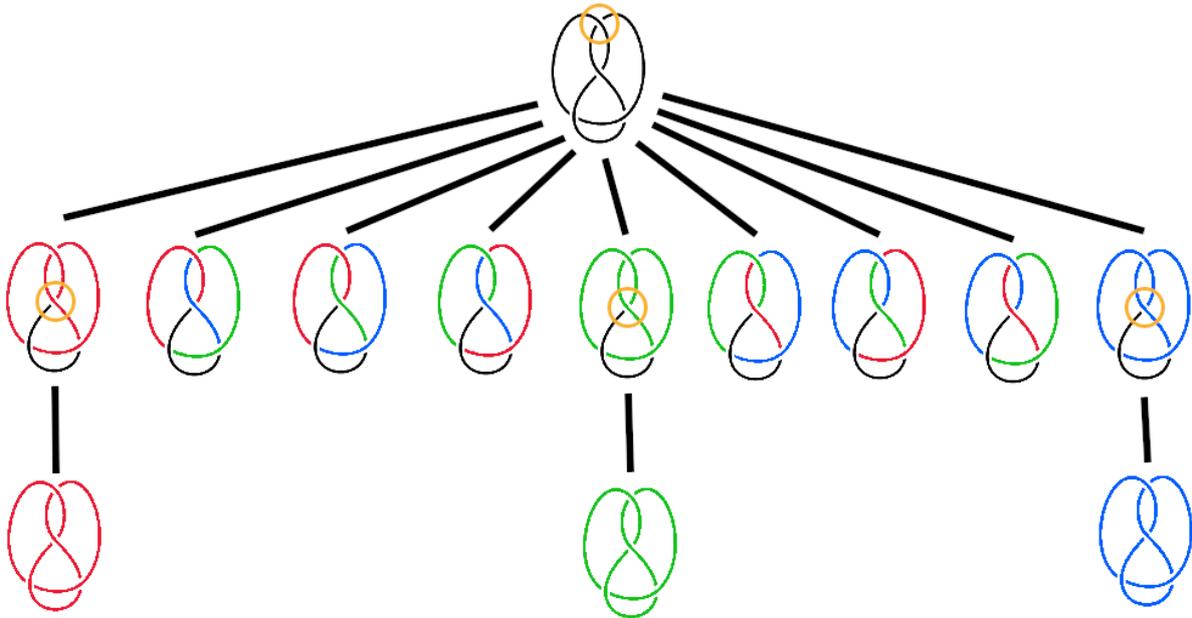


Abbildung 16.: Backtracking-Verfahren für den Achterknoten.

Bei vier Kreuzungen gibt es insgesamt $9^4 = 6561$ Färbungen, wir mussten allerdings nur 12 überprüfen. Wir konnten nämlich die Tatsache ausnutzen, dass nach dem Färben der ersten Kreuzung schon 6 der 9 Färbungen zum Scheitern verurteilt waren. Wir mussten also nur noch die übrigen 3 weiter auswerten.

Nun gilt es, die Idee zu verallgemeinern. Angenommen, wir bekommen eine Liste von Kreuzungen eines Knotens vorgelegt. Wir probieren als erstes alle Färbungen der ersten Kreuzung. Für jede der Möglichkeiten probieren wir dann alle Färbungen der zweiten Kreuzung und löschen diejenigen Färbungen, die sich in einem bestimmten Bogen uneinig mit der Färbung der ersten Kreuzung sind. Dann gehen wir für jede der verbleibenden Möglichkeiten alle Färbungen der dritten Kreuzung durch, löschen alle, die sich in einem Bogen uneinig sind und so weiter. Nach der letzten Kreuzung haben wir nur noch korrekte Färbungen, inklusive einfarbige.

Heuristik Backtracking ist schneller als bloßes Durchprobieren, weil wir die Kreuzungen in einer Reihenfolge anmalen, bei der wir früh erkennen, ob eine Färbung etwas taugt oder nicht. Es ist also essenziell für die Performanz, die Kreuzungen in einer geeigneten Reihenfolge durchzugehen.

Eine Reihenfolge ist dann zu bevorzugen, wenn es für die Färbung der Kreuzungen schon bald wenige Freiheitsgrade gibt. Eine Kreuzung hat dann wenige Freiheitsgrade, wenn möglichst viele ihrer Bögen schon gefärbt sind.

Angenommen, wir beginnen bei unserer Reihenfolge mit einer beliebigen Kreuzung. Die zweite Kreuzung sollte möglichst viele Bögen mit der ersten gemeinsam haben. Die dritte Kreuzung sollte möglichst viele Bögen mit den ersten beiden Kreuzungen gemeinsam haben und so weiter. Das gibt uns ein Verfahren, mit dem wir relativ schnell eine relativ gute Reihenfolge bekommen.

5.2. Divide-and-Conquer Algorithmus

Eine andere gängige Klasse von Verfahren für derartige algorithmisch lösbare Probleme sind so genannte Divide-and-Conquer Algorithmen. Dabei wird ein Problem gelöst, indem es immer weiter in Teilprobleme zerlegt wird, deren Lösungen am Ende wieder zusammengefügt werden.

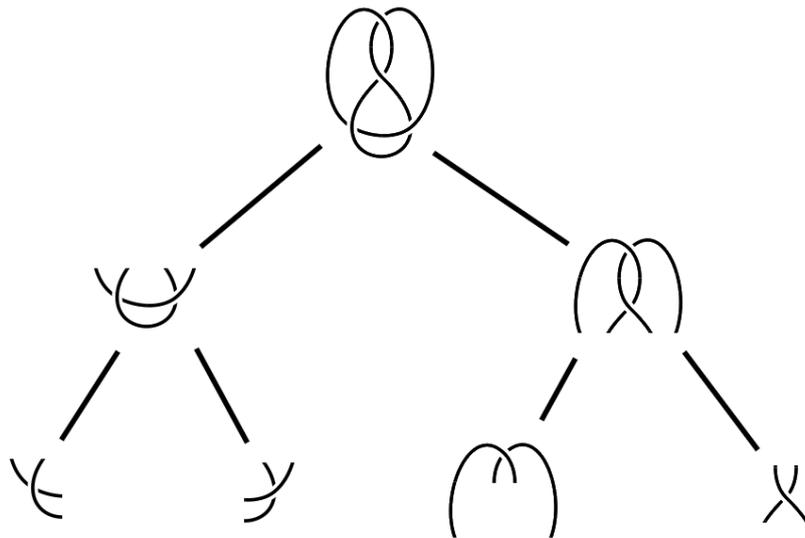


Abbildung 17.: Aufteilung des Achterknoten.

In unserem Fall bedeutet das folgendes: Zuerst teilen wir die Kreuzungen des Knoten in zwei Mengen auf. Dann finden wir für diese beiden Kreuzungsmengen alle erlaubten Färbungen. Dies können wir erreichen, indem wir die Kreuzungsmengen weiter aufteilen, bis wir bei einzelnen Kreuzungen landen. Dann bauen wir den Knoten wieder zusammen, indem wir jeweils die Färbungen von zwei Kreuzungsmengen kombinieren.

Um die Färbungen von zwei Kreuzungsmengen zu kombinieren, nehmen wir alle Möglichkeiten, eine Färbung aus der ersten Kreuzungsmenge und eine aus der zweiten zu wählen. Für jede dieser Möglichkeiten überprüfen wir dann, ob sich die beiden Färbungen an irgendeiner

5. Algorithmen zur Knotenfärbung

Kreuzung uneinig sind. Wenn sich die beiden Färbungen einig sind, können wir sie zu einer Färbung zusammenfassen. Diese zusammengefassten Färbungen ergeben dann die Menge der Färbungen der kombinierten Kreuzungsmengen.⁶⁰ In Abbildung 18 werden beispielhaft zwei Teile des Achterknoten kombiniert, indem unter allen möglichen Paaren an Färbungen diejenigen ausgewählt werden, die sich an den beiden Verbindungsbögen einig sind.

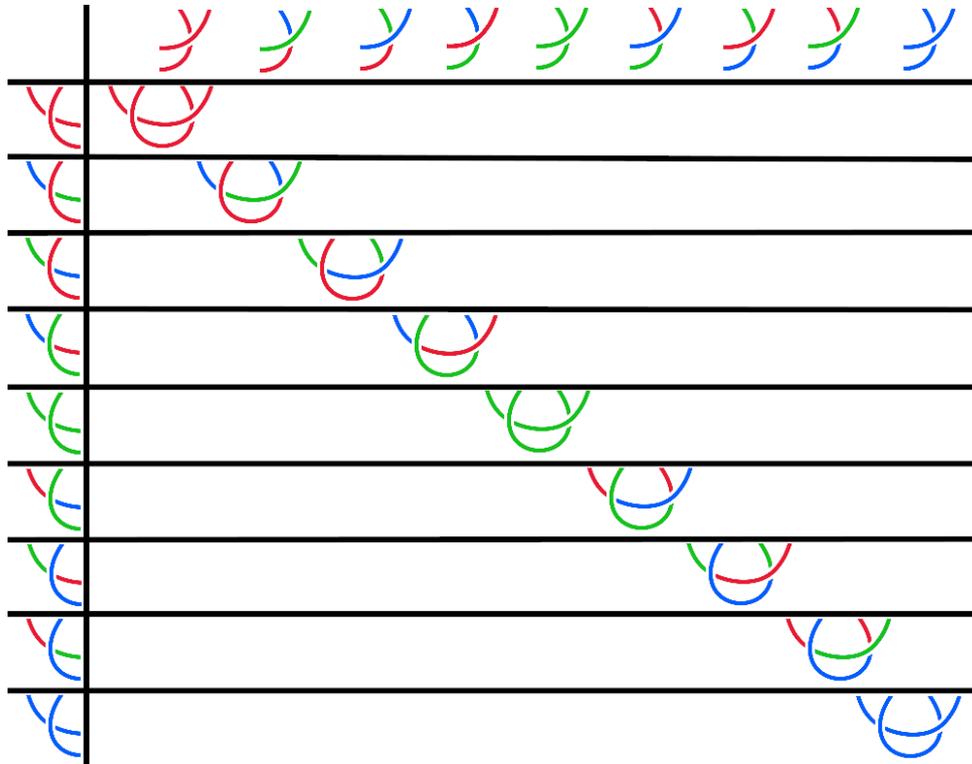


Abbildung 18.: Kombination zweier Kreuzungsmengen.

Offenbar ist es bei diesem Algorithmus von großer Bedeutung, wie genau der Knoten zerlegt wird. Ich habe mich bei meinem Algorithmus dafür entschieden, das Problem aus der anderen Richtung anzugehen und den Knoten nach und nach aus kleineren Teilen zusammenzubauen. So gesehen ist die Bezeichnung Divide-and-Conquer vielleicht unpassend. Wir starten also mit den kleinsten Teilen des Knotens – den Kreuzungen – und bauen den Knoten auf, indem wir Kreuzungsmengen nach dem oben beschriebenen Verfahren kombinieren. Ähnlich wie beim Backtracking-Verfahren ist es wichtig, schnell unmögliche Färbungen zu erkennen und auszusortieren. Daher sollte man also stets die Kreuzungsmengen kombinieren, bei deren Kombination am Ende eine möglichst kleine Zahl an Färbungen übrig bleibt. Es gilt also abzuschätzen, wie viele Färbungen eine Kombination zweier Kreuzungsmengen haben wird.

⁶⁰Wenn die beiden Kreuzungsmengen beide n Färbungen haben, braucht man mit diesem Verfahren $O(n^2)$ Schritte, um sie zu kombinieren. In meinem tatsächlichen Algorithmus verwende ich ein schnelleres Kombinationsverfahren, dessen Erklärung hier allerdings zu weit führen würde.

5. Algorithmen zur Knotenfärbung

Heuristik Wenn unsere kombinierte Kreuzungsmenge b Bögen und k Kreuzungen hat, und jeder der Bögen eine von c Farben annehmen kann, sollte die Anzahl der Färbungen in etwa c^{b-k} sein, da jeder Bogen c Färbungen hat (also insgesamt c^b Färbungen) und jede Kreuzung in etwa $\frac{1}{c}$ davon filtert, weil sie die Gleichung $a \triangleright b = c$ nicht erfüllen. Beispielsweise könnte man eine Kreuzung auf $3 * 3 * 3 = 27$ Arten färben, aber nur 9 davon, also $\frac{1}{3}$, sind im Sinne der Dreifärbbarkeit korrekt. Unsere Abschätzung für die Anzahl der Färbungen ist also c^{b-k} .

Mit dieser Abschätzung können wir also folgendermaßen vorgehen: Wir starten mit einzelnen Kreuzungen, und kombinieren dann immer die Kreuzungsmengen, für deren Kombination die Abschätzung c^{b-k} am kleinsten ist. Wir kombinieren so lange Kreuzungsmengen, bis wir sie alle zu einem Knoten kombiniert haben.

6. Performanz und Korrektheit der Algorithmen

Nachdem wir im letzten Kapitel zwei Verfahren zur Auffindung von Färbungen entwickelt haben, geht es in diesem Kapitel darum, diese auf Korrektheit und Effizienz zu analysieren. Zuerst werden konkrete Forschungsfragen gestellt. Im nächsten Schritt wird der Ursprung der Testdaten besprochen. Im Anschluss werden die Ergebnisse präsentiert und erläutert.

6.1. Forschungsfragen

Ich habe die beiden beschriebenen Algorithmen in der Programmiersprache *Haskell*⁶¹ implementiert und im Anhang hinzugefügt.

Ich werde auf den Code nicht weiter eingehen, da er dokumentiert ist und im Grunde genau das tut, was im letzten Kapitel beschrieben wurde. Es ist vorerst nicht klar, ob die Algorithmen tatsächlich korrekte Ergebnisse produzieren, da wir diesbezüglich auf Intuition vertraut haben. Die wichtigste Frage ist also die nach der Korrektheit der Verfahren:

Liefen die Algorithmen korrekte Ergebnisse?

Da beide Algorithmen dazu entwickelt wurden, das Färbungsproblem möglichst schnell zu lösen, stellt sich die Frage, ob sie das auch tun. Insbesondere ist es entscheidend, wie sich die Berechnungszeit verändert, wenn man die Größe der Eingabedaten erhöht. Daraus ergeben sich zwei Fragen:

Wie hängt die Berechnungszeit der Algorithmen von der Anzahl der Bögen im Knoten ab?

Wie hängt die Berechnungszeit der Algorithmen von der Anzahl der Farben ab?

⁶¹*Haskell Language* (2019). URL: <https://www.haskell.org/> (besucht am 29. 12. 2019).

6.2. Auswertung

Suche eines Datasets Um die Korrektheit der Programme zu überprüfen, sind Testdaten notwendig, also eine Liste von Knoten und Quandles, und dazu noch für jedes Paar an Knoten und Quandles die Information, ob der Knoten färbbar ist. Eine gute Wahl sind die Daten aus einem Paper von Andrew Fish, Alexei Lisitsa und David Stanovský⁶². Die Daten der im Paper beschriebenen Berechnungen sind auf der Homepage von David Stanovský verfügbar.⁶³ Die downloadbaren Files beinhalten eine Liste von 1288 Knoten mit 12 Kreuzungen und einige sogenannte Torusknoten.

Die Knoten sind gegeben als einzelne **.knot** Files, wobei ein solches File in jeder Zeile eine Kreuzung beinhaltet. Dabei werden die Indizes des überkreuzenden, linken und rechten Bogens nacheinander hingeschrieben. Der Trefoil wäre also folgendermaßen notiert:

```
1 2 3
3 1 2
2 3 1
```

Die Quandles sind alle im File **simplequandles.dat** nacheinander über die Wertetabelle von ▷ gegeben. Dazu gibt es auch noch ein Programm, das Färbbarkeit berechnet. Das Verfahren der Autoren dieses Papers reduziert das Färbungsproblem auf das sogenannte Boolean Satisfiability Problem (SAT). Für SAT gibt es sehr gut optimierte Algorithmen, was man sich in dem Fall zunutze machen kann.

Überprüfung der Korrektheit anhand von Testfällen Um Zeit zu sparen, habe ich für die Überprüfung der Korrektheit nicht den gesamten Datensatz von David Stanovský verwendet. Für Knoten habe ich die ersten 100 der 12-Kreuzungs-Knoten verwendet und für die Quandles die ersten 10. Als Testmenge habe ich dann alle Paare von diesen Knoten und Quandles verwendet. Der Test war für beide implementierten Algorithmen erfolgreich, sie stimmen also mit den Resultaten von Stanovský überein.

Analyse der Performanz Die Performanz der Algorithmen hängt von zwei Größen ab, der Größe des Quandles und der Größe des Knoten. Um die Performanz in Abhängigkeit der Quandlegröße zu messen wurde als Knotenmenge eine Menge von 25 zufällig generierten Knoten der Größe 9 verwendet. Diese Knoten wurden eigens dafür durch ein Programm von mir generiert. Es wurden für die Quandlegrößen 5, 7, 9, 11, 13, 15, 17 jeweils zwei Quandles genommen und zu einer Testmenge von 50 Knoten-Quandle Paaren vereint. Es gab also insgesamt 7 Testmengen

⁶²Fish, Lisitsa und Stanovský (2015).

⁶³Stanovský (2019).

6. Performanz und Korrektheit der Algorithmen

mit jeweils 50 Elementen, die für beide Algorithmen getestet wurden. Zum Testen habe ich die Haskell-Library *Criterion* verwendet, welche die Algorithmen für jede der Testmengen mehrmals ausgeführt hat, um externe Störfaktoren zu eliminieren. Die Berechnungen wurden auf einem Intel Core 2 Duo Prozessor (E6600, 2.4 GHz) durchgeführt. Das Ergebnis des Profilings nach Quandlegröße ist in Abbildung 19 zu sehen. Im Balkendiagramm werden die verschiedenen Knoten- und Quandlegrößen in Bezug auf die Berechnungsdauer der beiden Algorithmen dargestellt. Die Berechnungsdauer wird in Sekunden gemessen, wobei zu beachten ist, dass die in der Grafik vorkommenden Zahlen noch durch 50, also die Anzahl an Knoten-Quandle Paaren pro Testmenge dividiert werden muss, um die durchschnittliche Berechnungsdauer für eine bestimmte Quandlegröße zu erhalten.

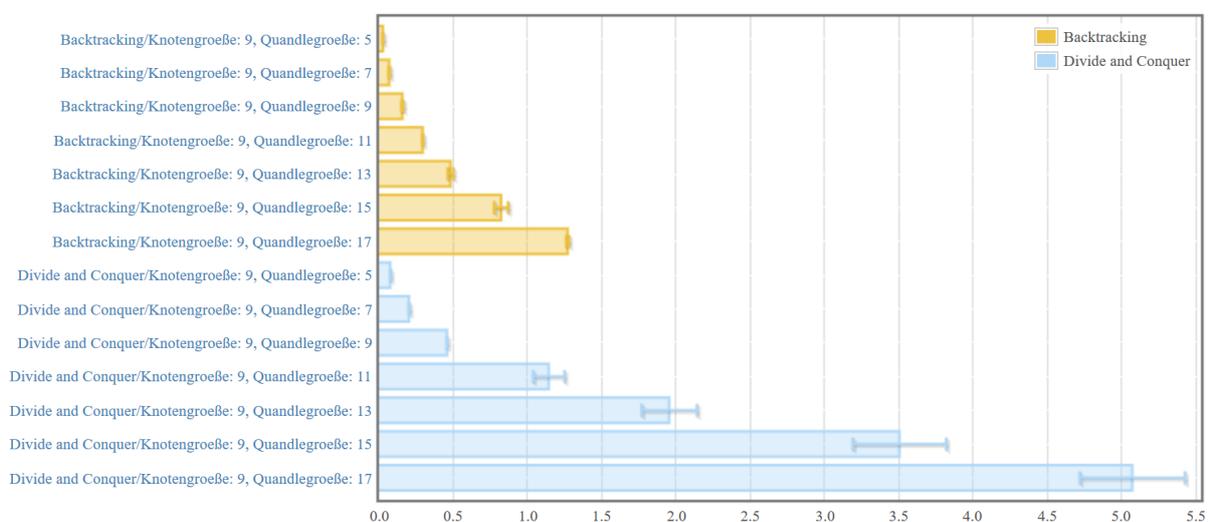


Abbildung 19.: Profiling nach Quandlegröße (Zeit in Sekunden).

Die Performanz hängt außerdem noch von der Größe des Knoten ab. Hierbei wurden als Quandlemenge wiederum zwei Quandles der Größe 9 verwendet und für die Knotengrößen 5, 7, 9, 11, 13, 15, 17 jeweils 25 Knoten zufällig generiert. Es waren also wieder 7 Testmengen mit jeweils 50 Elementen. Das Ergebnis ist in Abbildung 20 zu sehen.

Um die Abhängigkeit der Berechnungsdauer von sowohl Knotengröße als auch Quandlegröße zu erkennen, wurde zusätzlich noch ein Profiling mit jeweils 2 Quandles und 25 Knoten der Größen 5, 7, 9, 11, 13 durchgeführt. Hier gab es also 5 Testmengen mit jeweils 50 Knoten-Quandle Paaren. Für die Größe 15 dauerte das Profiling des Divide and Conquer Algorithmus zu lange, weshalb ich bei der Größe 13 (für jeweils Knoten und Quandle) aufhörte. Das Ergebnis ist in Abbildung 21 zu sehen.

6. Performanz und Korrektheit der Algorithmen

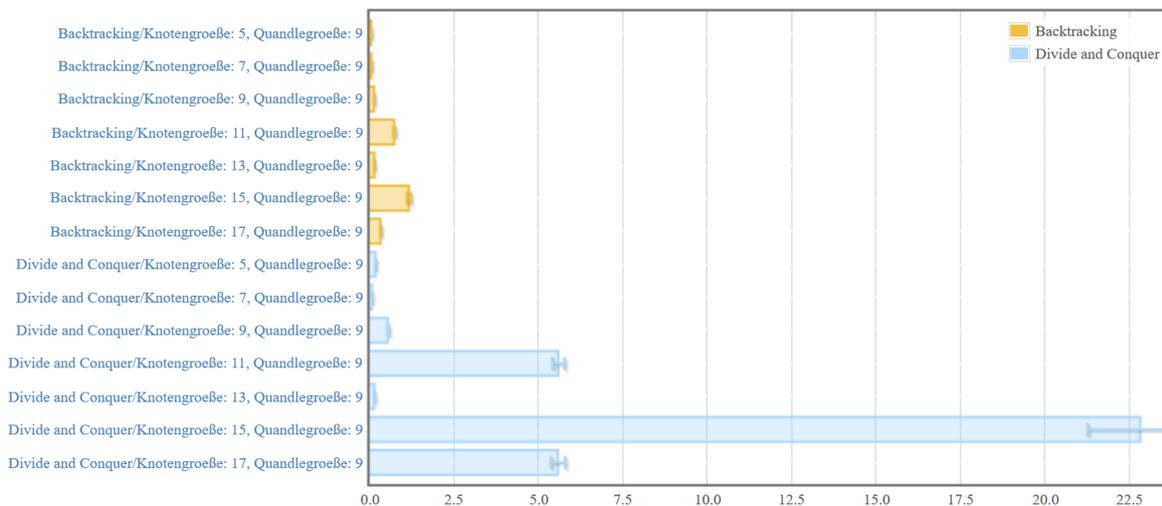


Abbildung 20.: Profiling nach Knotengröße (Zeit in Sekunden).

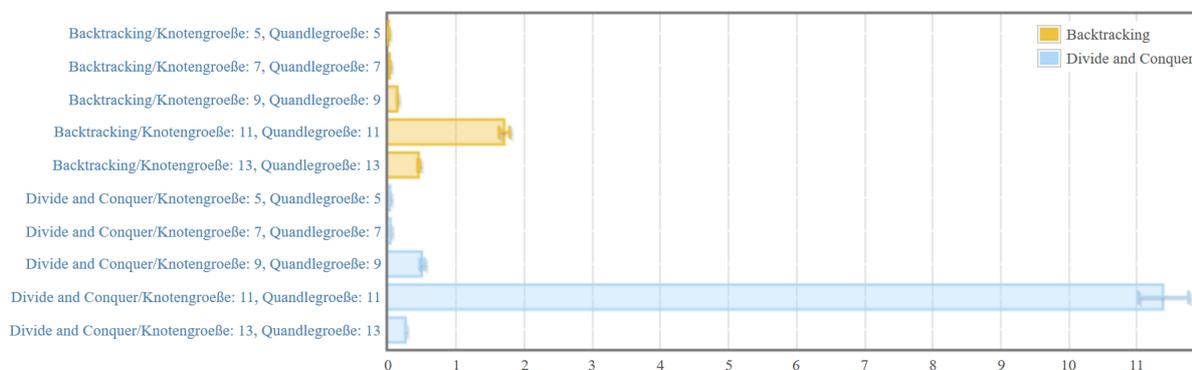


Abbildung 21.: Profiling nach Knotengröße und Quandlegroße (Zeit in Sekunden).

6.3. Diskussion

In beiden Fällen ist das Backtracking-Verfahren schneller als der Divide and Conquer Algorithmus. Dies könnte daran liegen, dass die heuristische Suche beim Backtracking bessere Ergebnisse liefert. Es wäre aber auch denkbar, dass der Divide-and-Conquer Algorithmus sein Potential erst bei großen Knoten voll ausschöpfen kann und bei relativ kleinen Knoten der Overhead, der durch das komplexere Verfahren entsteht, noch überwiegt.

Spannend ist, dass die Komplexität beider Algorithmen viel sensibler auf die Quandlegroße reagiert als auf die Knotengröße. Größere Knoten sind also tendenziell leichter als größere Quandles. In den Profiling-Ergebnissen von Stanovský ergibt sich ein ähnliches Bild. Auf der anderen Seite hätte ein Brute-Force Verfahren, das gewissermaßen zufällig Färbungen durchprobiert, bei q Farben und k Bögen eine Komplexität von circa $q^k = e^{k \cdot \ln(q)}$, was offensichtlich viel stärker von der Größe des Knotens k abhängt.

6. Performanz und Korrektheit der Algorithmen

Das ließe sich dadurch erklären, dass wir unsere Bemühungen, die Effizienz zu steigern, hauptsächlich darauf verwendet haben, Bögen in einer besseren Anordnung durchzuprobieren und nicht darauf, Farben geeignet auszuwählen. Wir haben in unseren Algorithmen einen Quandle wie eine Black Box behandelt, die bestimmte Kreuzungen erlaubt und andere nicht. Stanovský's Reduktion des Problems auf SAT geht auch nicht auf die innere Struktur der Quandles ein. Eventuell wäre es mit ausgeklügelteren Algorithmen möglich, die Quandleaxiome dafür zu verwenden, das Durchprobieren von Farben in einer geeigneteren Reihenfolge auszuführen.

Interessant ist auch, dass die Berechnungsdauer viel stabiler auf unterschiedliche Quandlegrößen reagiert als auf unterschiedliche Knotengrößen, soll heißen, die Entwicklung der Berechnungsdauer nach Quandles in Abbildung 19 ist sehr viel balancierter als die Entwicklung nach Knoten in Abbildung 20. In den Testmengen kommen 25 verschiedene Knoten und nur 2 verschiedene Quandles vor. Man würde eigentlich erwarten, dass sich durch die vielen Knoten eventuelle Schwankungen ausgleichen und die Komplexität stärker von der Quandlegröße abhängig ist. Das war allerdings nicht der Fall. Vielmehr zeigt sich bei der Quandlegröße ein klarer Trend, bei der Knotengröße fällt es dagegen schon schwer, eine klar steigende Tendenz zu erkennen. Andererseits brauchen beide Algorithmen bei den Knotengrößen 11 und 15 am längsten. Das heißt, obwohl die Abhängigkeit der Effizienz von der Knotengröße zufällig erscheint, scheint sie für die beiden Algorithmen doch zu korrelieren. Es scheint also so zu sein, dass es für Knoten bestimmter Größen im Allgemeinen schwer ist, Färbungen zu finden.

Ich habe viele verschiedene Profilings mit unterschiedlichen Implementierungen der Algorithmen und verschiedenen Arten von Testmengen durchgeführt und es ergab sich kaum ein kohärentes Bild, weswegen die gerade getätigten Vermutungen auch nicht empirisch belegt sind. Beispielsweise unterscheiden sich verschiedene Implementierungen des Backtracking Algorithmus in ihrer Effizienz circa um einen Faktor 7, was die Frage aufwirft, wie gut man die derzeitige Implementierung noch optimieren könnte. Die Art von Knoten, die David Stanovský in seinem Paper zum Profiling nach Knotengröße verwendet, so genannte Torusknoten, führten bei meinen Algorithmen zu viel stabileren und kürzeren Berechnungsdauern als komplett zufällige Knoten. Würde man die Implementierung des Knoten-Zufallsgenerators also ändern, könnte das Bild womöglich anders aussehen.

Meine Algorithmen waren gerade bei großen Quandles beide nicht im Stande, die Effizienz von Stanovský's Verfahren zu übertreffen. Das ist wenig überraschend, wenn man bedenkt, dass in seinem Verfahren das Problem auf SAT reduziert wird, ein Problem, das seit Jahrzehnten von zahllosen Forschern untersucht wird. Meine Verfahren haben jedoch für die hier diskutierten Knoten- und Quandlegrößen Berechnungsdauern in der Größenordnung von $10ms$ bis $2s$, was für die Anwendung in diesem Bereich vollkommen akzeptabel ist.

7. Zusammenfassung und Ausblick

Wir haben in dieser Einführung verschiedene Grundkonzepte aus der Knotentheorie kennengelernt und uns dann auf Färbungen spezialisiert. Nach einer Hierarchie von verschiedenen Zugängen zur Idee von Färbungen landeten wir schließlich bei Quandles, die alle anderen Färbungsarten verallgemeinern. Es wurden Algorithmen für das Färbungsproblem entwickelt und deren Effizienz anhand von verschiedenen Kriterien getestet.

In einer vorwissenschaftlichen Arbeit muss man sich fokussieren, um nicht nur an der Oberfläche zu kratzen. Daher habe ich mich entschieden, mich auf Färbungen zu spezialisieren. Dadurch wurden natürlich viele Themen vernachlässigt, wie zum Beispiel Polynomvarianten, die Artin'sche Zopfgruppe, die Yang-Baxter-Gleichungen und der Zusammenhang zwischen Knoten und Feynman-Diagrammen, Knoten in höherdimensionalen Räumen, topologische Interpretationen von Färbung, Quandlecohomologie und vieles mehr. Dadurch sollte nicht der Eindruck entstehen, dass Färbungen im Zentrum der Knotentheorie stehen, im Gegenteil, der Großteil der Literatur erwähnt Färbungen nur am Rande. Die Literatur selbst ist ebenfalls relativ divers in ihren Spezialisierungen und Notationen, was vielleicht daran liegen könnte, dass die Knotentheorie noch viel jünger ist als große mathematische Theorien wie Algebra oder Analysis.

Danksagung

Ich möchte mich noch bei allen Menschen bedanken, die mir durch Tipps und Vorschläge beim Schreiben der Arbeit geholfen haben. Dr. Georg Grasegger, der die Planung des Programms *Young Scientists der Mathematik* leitete, als ich im Wintersemester 2018/2019 teilnahm, gab mir den wertvollen Tipp, *Knotentheorie für Einsteiger* von Charles Livingston zu lesen, was mich schließlich dazu motivierte, Färbungen zum Fokus der Arbeit zu machen. a. Univ.-Prof. Dr. Herbert Prähofer motivierte mich, ein Backtracking-Verfahren zu entwickeln, nachdem ich den Divide-and-Conquer Algorithmus schon entwickelt hatte, und die beiden Algorithmen zu vergleichen. Assoz. Univ.-Prof. Dr. Erhard Aichinger gab mir den Hinweis auf die Arbeiten von David Stanovský, die mir als Testdaten zum Validieren meines Verfahrens sehr halfen. Ich möchte mich bei meinem Vater dafür bedanken, dass er mich mit den letzten beiden Genannten in Verbindung gebracht hat, mich bei der Planung des Inhalts unterstützt hat und mir bei der Arbeit mit \LaTeX geholfen hat. Schließlich möchte ich mich bei meinem Betreuungslehrer Mag. Johannes Teuschl bedanken, dass er mir von sich aus anbot, mein Thema zu übernehmen, meinen Enthusiasmus über Knotentheorie teilen konnte und mir dabei half, die Hürden zu überwinden, die bei einem eher unüblichen Thema unweigerlich entstehen.

Anhang A.

Programmcode

Backtracking.hs

```
1 module Backtracking where
2   import Data.Maybe (maybeToList)
3   import Data.List as List
4   import Data.Map as Map
5   import Data.Ord (comparing)
6   import Data.Set as Set
7   import Datatypes (Quandle, Knot, Coloring, Arc, Color)
8
9
10  colorings :: Quandle -> Knot -> [Coloring]
11  colorings quandle knot = let
12    crossings = bestCrossingOrder Set.empty $ Set.fromList knot
13    in coloringsHelper quandle crossings Map.empty
14  {-
15   main function of this module
16   calculates a List of all 'legal' Colorings of
17   a given Quandle and Knot
18 -}
19
20
21  bestCrossingOrder :: Set (Arc, Arc, Arc) -> Set (Arc, Arc, Arc) -> [(Arc, Arc, Arc)]
22  bestCrossingOrder resolved unresolved = if Set.null unresolved then [] else
23    let start = bestCrossing resolved unresolved
24        rest = bestCrossingOrder (Set.insert start resolved) (Set.delete start unresolved)
25    in start : rest
26  {-
27   calculates the best order at which to evaluate the crossings,
28   given Sets of unresolved and resolved crossings
29 -}
30
31
```

Anhang A. Programmcode

```
32 bestCrossing :: Set (Arc, Arc, Arc) -> Set (Arc, Arc, Arc) -> (Arc, Arc, Arc)
33 bestCrossing resolved unresolved = List.maximumBy usingHeuristic unresolvedArcs where
34   usingHeuristic = comparing heuristic
35   unresolvedArcs = Set.toList unresolved
36   resolvedArcs = Set.unions $ toSet <$> Set.toList resolved
37   toSet (x, y, z) = Set.fromList [x, y, z]
38   heuristic crossing = Set.size $ Set.intersection resolvedArcs $ toSet crossing
39 {-
40   calculates the best suited crossing to evaluate next,
41   given Sets of unresolved and resolved crossings
42 -}
43
44
45 coloringsHelper :: Quandle -> Knot -> Coloring -> [Coloring]
46 coloringsHelper _ [] partial = [partial]
47 coloringsHelper quandle (k:ks) partial = do
48   possibility <- quandle
49   applied <- maybeToList $ applyCrossing k possibility partial
50   coloringsHelper quandle ks applied
51 {-
52   takes a Quandle and a (partially evaluated) Knot and a partially evaluated Coloring and
53   returns a List of 'legal' ways to complete this coloring
54 -}
55
56
57 applyCrossing :: (Arc, Arc, Arc) -> (Color, Color, Color) -> Coloring -> Maybe Coloring
58 applyCrossing (i1, i2, i3) (c1, c2, c3) partial = do
59   s1 <- apply i1 c1 partial
60   s2 <- apply i2 c2 s1
61   apply i3 c3 s2
62 {-
63   tries to color a crossing in a partially evaluated Coloring with specific Colors.
64   returns Nothing if the partially evaluated Coloring
65   contains different Colors for the relevant arcs
66 -}
67
68
69 apply :: Arc -> Color -> Coloring -> Maybe Coloring
70 apply arcIndex color partial = case Map.lookup arcIndex partial of
71   Nothing -> Just $ Map.insert arcIndex color partial
72   Just x -> if x == color then Just partial else Nothing
73 {-
74   tries to apply a Color to an Arc in a partially evaluated Coloring.
75   returns Nothing if the Arc is already colored differently
76 -}
```

```

1  module DivideAndConquer where
2      import Datatypes (Quandle, Knot, Coloring, Arc, Color)
3      import Data.List as List
4      import Data.Map.Strict as Map
5      import Data.Maybe (maybeToList)
6      import Data.Set (fromList)
7      import Data.Ord (comparing)
8      import Control.Monad (guard)
9
10     data MergingTree = Leaf [Arc] | Tree MergingTree MergingTree [Arc]
11     {-
12     a MergingTree contains the intended merging structure for a part of a knot.
13     It can be one of two things:
14         - a Leaf i.e. a crossing, containing a List of three Arcs.
15         - a pair of MergingTrees which are going to be merged later on,
16           together with a list of their Arcs.
17     -}
18
19
20     instance Eq MergingTree where
21         Leaf a == Leaf b = a == b
22         Leaf _ == Tree _ _ _ = False
23         Tree _ _ _ == Leaf _ = False
24         Tree _ _ a == Tree _ _ b = a == b
25
26
27     colorings :: Quandle -> Knot -> [Coloring]
28     colorings quandle knot = let
29         tree = createMergingTree knot
30         in evaluate tree quandle
31     {-
32     main function of this module
33     calculates a List of all 'legal' Colorings of
34     a given Quandle and Knot
35     -}
36
37
38     createMergingTree :: Knot -> MergingTree
39     createMergingTree knot = head $ (iterate' mergeBest $ toTree <$> knot) !! n where
40         n = (length knot) - 1
41         toTree (x, y, z) = Leaf [x, y, z]
42     {-
43     creates a MergingTree from a Knot by starting with Leaf's
44     and always merging the two best suited MergingTrees into a larger one
45     -}

```

Anhang A. Programmcode

```

46
47
48 mergeBest :: [MergingTree] -> [MergingTree]
49 mergeBest trees = (merge a b) : (List.filter (\x-> x/=a && x/=b) trees) where
50     (a, b) = minimumBy (comparing heuristic) treePairs
51     treePairs = [(x, y) | x <- trees, y <- trees, x /= y]
52     --merges the two best suited MergingTrees in a List of MergingTrees
53
54
55 heuristic :: (MergingTree, MergingTree) -> Int
56 heuristic (a, b) = la + lb - li - ca - cb where
57     la = length $ arcs a
58     lb = length $ arcs b
59     li = length $ intersect (arcs a) (arcs b)
60     ca = crossings a
61     cb = crossings b
62     --determines how well a pair of MergingTrees is suited for merging
63
64
65 arcs :: MergingTree -> [Arc]
66 arcs (Tree _ _ a) = a
67 arcs (Leaf a) = a
68     --returns a List of all Arcs in the MergingTree
69
70
71 crossings :: MergingTree -> Int
72 crossings (Leaf _) = 1
73 crossings (Tree a b _) = crossings a + crossings b
74     --returns the number of crossings a MergingTree contains
75
76
77 merge :: MergingTree -> MergingTree -> MergingTree
78 merge a b = Tree a b (List.nub $ arcs a ++ arcs b)
79 {-
80 merges two MergingTrees.
81 However, the combination of the coloring sets doesn't take place here.
82 -}
83
84
85 evaluate :: MergingTree -> Quandle -> [Coloring]
86 evaluate (Leaf [x, y, z]) quandle = let
87     fromQuandleElement (a, b, c) = [(x, a), (y, b), (z, c)]
88     in Map.fromList <$> fromQuandleElement <$> quandle
89 evaluate (Tree a b _) quandle = let
90     evalA = evaluate a quandle
91     evalB = evaluate b quandle

```

Anhang A. Programmcode

```
92     arcIntersection = List.intersect (arcs a) (arcs b)
93     in combine evalA evalB arcIntersection
94 {-
95     evaluates all 'legal' colorings, given a MergingTree and a quandle
96     -}
97
98
99 combine :: [Coloring] -> [Coloring] -> [Arc] -> [Coloring]
100 combine x y border = do
101     let pack t = packMapBy t border
102         (boxVals, coloringsX) <- Map.toList $ pack x
103         coloringsY <- maybeToList $ Map.lookup boxVals $ pack y
104         ax <- coloringsX
105         ay <- coloringsY
106         return $ Map.union ax ay
107 {-
108     combines two Lists of Colorings into one List of combined Colorings,
109     given additionally a List of Arcs which they are supposed color equally.
110     -}
111
112
113 packMapBy :: [Coloring] -> [Arc] -> Map.Map [Color] [Coloring]
114 packMapBy colorings keyVars = List.foldl' insertColoring Map.empty colorings where
115     insertColoring :: Map.Map [Color] [Coloring] -> Coloring -> Map.Map [Color] [Coloring]
116     insertColoring resultingMap coloring = if Map.member valuesAtKeyVars resultingMap
117         then Map.adjust appendColoring valuesAtKeyVars resultingMap
118         else Map.insert valuesAtKeyVars [coloring] resultingMap where
119         valuesAtKeyVars = List.map (\k -> coloring Map.! k) keyVars
120         appendColoring list = coloring : list
121     --sorts a given List of Colorings by the colors they have on a given List of Arcs
```

Datatypes.hs

```
1 module Datatypes where
2     import Data.Map as Map
3
4     type Color = Int
5
6     type Arc = Int
7
8     type Quandle = [(Color, Color, Color)]
9     {-
10     the List contains all the accepted colorings at a crossing of the knot,
11     in terms of quandles:  $y \mid x = z$  is equivalent to  $(x,y,z) \text{ `elem` } q :: \text{Quandle}$ 
12     the order might be somewhat unusual and originates from David Stanovský.
13     -}
14
```

Anhang A. Programmcode

```
15  type Coloring = Map.Map Arc Color
16  --a Coloring maps Arcs to Colors
17
18  type Knot = [(Arc, Arc, Arc)]
19  {-
20  contains the crossings of a given knot,
21  where (x,y,z) would indicate that arc x splits y to the right and z to the left.
22  -}
```

Literatur

- Bar-Natan, Dror (2019). *KnotTheory*'. [http://katlas.org/wiki/The_Mathematica_Package_KnotTheory'](http://katlas.org/wiki/The_Mathematica_Package_KnotTheory). (Besucht am 30. 10. 2019).
- Cha, Jae Choon und Charles Livingston (2019). *KnotInfo: Table of Knot Invariants*. <http://www.indiana.edu/~knotinfo>. (Besucht am 07. 12. 2019).
- Clark, W. E. u. a. (2014). „Quandle colorings of knots and applications“. In: *Journal of Knot Theory and Its Ramifications* 23 (6). DOI: 10.1142/S0218216514500357.
- contributors, Wikipedia (2015). *Wild knot — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Wild_knot&oldid=679495888. (Besucht am 30. 10. 2019).
- contributors, Wikipedia (2017). *Tricolorability — Wikipedia, The Free Encyclopedia*. <https://en.wikipedia.org/w/index.php?title=Tricolorability&oldid=786521874>. (Besucht am 30. 10. 2019).
- contributors, Wikipedia (2019). *Knot theory — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Knot_theory&oldid=925007816. (Besucht am 30. 10. 2019).
- Feichtner, Eva Maria (1994). „Polynom invarianten der Knotentheorie – Quellen ihrer Definition“. Magisterarb. FU Berlin.
- Fenn, Roger und Colin Rourke (1992). „Racks and links in codimension two“. In: *Journal of Knot Theory and Its Ramifications* 1.4, S. 343–406.
- Fischer, Gerd (2005). *Lineare Algebra*. Springer Spektrum. ISBN: 978-3-658-03944-8.
- Fish, Andrew, Alexei Lisitsa und David Stanovský (2015). „A Combinatorial Approach to Knot Recognition“. In: *Embracing Global Computing in Emerging Economies*. Hrsg. von Horne, Ross. Cham: Springer International Publishing, S. 64–78. ISBN: 978-3-319-25043-4.
- Freyd, P. u. a. (1985). „A new polynomial invariant of knots and links“. In: *Bulletin (New Series) of the American Mathematical Society* 12.2, S. 239–246.
- Gordon, C. McA. und J. Luecke (1989). „Knots are determined by their complements“. In: *Bulletin (New Series) of the American Mathematical Society* 20.1. [21], S. 83–87.
- Haskell Language* (2019). URL: <https://www.haskell.org/> (besucht am 29. 12. 2019).
- Jones, Vaughan F. R. (1985). „A polynomial invariant for knots via von Neumann algebras“. In: *Bulletin (New Series) of the American Mathematical Society* 12.1, S. 103–111.

- Joyce, David (1982). „A classifying invariant of knots, the knot quandle“. In: *Journal of Pure and Applied Algebra* 23.1, S. 37–65. ISSN: 0022-4049.
- Kauffman, Louis (1995). *Knoten – Diagramme, Zustandsmodelle, Polynomvarianten*. Heidelberg, Berlin, Oxford: Spektrum Akademischer Verlag. ISBN: 3860252321.
- Kauffman, Louis H. und Sofia Lambropoulou (2009). „Hard Unknots and Collapsing Tangles“. In: *Series on Knots and Everything – Introductory Lectures on Knot Theory*. Hrsg. von Kauffman, Louis H u. a. Bd. 46. World Scientific, S. 187–247. DOI: 10.1142/9789814313001_0009.
- Kuperberg, Greg (2014). „Knottedness is in NP, modulo GRH“. In: *Advances in Mathematics* 256, S. 493–506. ISSN: 0001-8708.
- Kussin, Dirk (2005). *Knoten und Verschlingungen*. http://math-www.uni-paderborn.de/~dirk/Vorlesungen/Knoten/skript_12c.pdf. (Besucht am 07. 12. 2019).
- Lackenby, Marc (2015). „A polynomial upper bound on Reidemeister moves“. In: *Annals of Mathematics* 182.2, S. 491–564. ISSN: 0003486X.
- Lackenby, Marc (2016a). „Elementary knot theory“. In: *arXiv e-prints*.
- Lackenby, Marc (2016b). „The efficient certification of knottedness and Thurston norm“. In: *arXiv e-prints*.
- Livingston, Charles (1995). *Knotentheorie für Einsteiger*. Vieweg Lehrbuch. Vieweg+Teubner Verlag. ISBN: 9783528066604.
- Menasco, William W. und Morwen B. Thistlethwaite (eds.) (2004). *Table of Contents for the Handbook of Knot Theory*.
- Przytyck, Józef H. (1998). „3-coloring and other elementary invariants of knots“. In: *Banach Center Publications* 42, S. 275–295. DOI: 10.4064/-42-1-275-295.
- Stanovský, David (2019). *Calculating with Quandles*. <http://www.karlin.mff.cuni.cz/~stanovsk/quandles/>. (Besucht am 30. 10. 2019).
- Stillwell, John (2012). *Classical Topology and Combinatorial Group Theory*. Graduate Texts in Mathematics. Springer New York.
- Whitten, Wilbur (1987). „Knot Complements and Groups“. In: *Topology* 26.1, S. 41–11.