

Planare Graphen und ihre Dualgraphen auf Zylinderoberflächen¹

Christopher Auer²

Abstract: Die Arbeit beschäftigt sich mit planaren Zeichnungen ungerichteter und gerichteter Graphen auf Zylinderoberflächen. Im ungerichteten Fall werden die Knoten auf einer Linie parallel zur Zylinderachse positioniert, während die Kanten diese Linie nicht schneiden. Es wird gezeigt, dass eine planare Zeichnung genau dann möglich ist, wenn die Kanten des Graphen in einer double-ended queue (Deque) verarbeitet werden können. Als Konsequenz ergibt sich, dass die Deque genau die planaren Graphen mit Hamiltonpfad charakterisiert. Dies erweitert die bereits bekannte Charakterisierung planarer Graphen mit Hamiltonkreis durch den Doppelstack. Im gerichteten Fall verlaufen die Kantenkurven entweder in Richtung der Zylinderachse (**SUP**) oder um die Achse herum (**RUP**). Die Arbeit charakterisiert **RUP**-Graphen und zeigt, dass **RUP** und **SUP** ihre Rollen tauschen, wenn man Graph und Dualgraph betrachtet. Mit Hilfe dieser Charakterisierung wird ein Erkennungs-Algorithmus für **RUP**-Graphen entwickelt.

1 Einführung

Eine planare Zeichnung eines Graphen weist den Knoten unterschiedliche Punkte in der Ebene zu und bildet die Kanten auf einfache Jordankurven ab, die keine gemeinsamen Punkte besitzen außer an gemeinsamen Knoten. Setzt man nun in die durch die Zeichnung abgegrenzten Regionen Flächenknoten und verbindet zwei benachbarte Flächenknoten durch eine Kante, so erhält man den Dualgraphen. Die vorliegende Arbeit betrachtet zwei spezielle Arten von Graph-Zeichnungen und deren Dualgraphen auf der 3D-Zylinderoberfläche. Zunächst werden linear-zyllindrische Zeichnungen verwendet um neue neue Einsichten in die Arbeitsweise fundamentaler Datenstrukturen wie dem Stack, der Queue und der Deque zu gewinnen (Kapitel 2). In diesen Zeichnungen werden die Knoten auf einer Linie platziert werden, die parallel zur Zylinderachse verläuft, während die Kanten diese Linie nicht schneiden dürfen. Im zweiten Teil der Arbeit werden rollend aufwärtsplanare Zeichnungen von gerichteten Graphen charakterisiert (Kapitel 3). Im diesem Fall müssen die Kantenkurven entweder in Richtung der Zylinderachse verlaufen („standing upward planar“, **SUP**) oder sich um die Achse herumbewegen („rolling upward planar“, **RUP**). Hier wird ein „Ampèresches Gesetz“ hergeleitet, das **SUP** und **RUP** über deren Dualgraphen in Beziehung setzt.

¹ Englischer Titel der Dissertation: „Planar Graphs and their Duals on Cylinder Surfaces“

² Universität Passau, auerc@fim.uni-passau.de

2 Linear-zylindrische Zeichnungen

2.1 Graph-Layouts in Datenstrukturen

Linear-zylindrische Zeichnungen spielen insbesondere im Zusammenhang mit Graph-Layouts in Datenstrukturen wie dem Stack, dem Doppelstack, der Queue und der Deque eine wichtige Rolle. In einem Graph-Layout werden die Kanten eines Graphen in einer Datenstruktur verarbeitet. Ein *lineares Layout* legt dabei die Verarbeitungsreihenfolge der Knoten fest. An jedem Knoten werden zuerst alle Kanten zu Vorgängern im linearen Layout aus der Datenstruktur entfernt und danach alle Kanten zu Nachfolgern in die Datenstruktur eingefügt. Dies muss nach den durch die Datenstruktur festgelegten Regeln geschehen wie z.B. „last in, first out“ (LIFO) bei einem Stack (Stapel) und „first in, first out“ (FIFO) bei einer Queue (Warteschlange).

Abbildung 1(a) zeigt ein *Stack-Layout*, wobei die gestrichelte Kante für den Moment ignoriert wird. Die Knoten sind auf der gepunkteten horizontalen Linie gemäß dem linearen Layout angeordnet und die Kanten sind als Bögen oberhalb gezeichnet. Der Inhalt des Stacks vor und nach der Verarbeitung des jeweiligen Knoten ist unterhalb dargestellt. An Knoten 2 wird die Kante e_1 vom Stack entfernt und die Kante e_5 wird eingefügt. Kante e_4 (gestrichelt) muss am Knoten 2 vor Kante e_5 auf den Stack gelegt werden. Der Inhalt des Stacks unmittelbar vor der Verarbeitung von Knoten 3 ist somit (e_5, e_4, e_2, e_3) (von oben nach unten zu lesen). An Knoten 3 kann Kante e_2 nicht vom Stack entfernt werden, da e_4 über ihr auf dem Stack liegt. Wie in Abb. 1(a) zu sehen ist, kreuzen sich e_2 und e_4 . Für ein gegebenes lineares Layout ist eine kreuzungsfreie Darstellung wie in Abb. 1(a) genau dann möglich wenn alle Kanten im Stack verarbeitet werden können. Dies entspricht einer gültigen Schachtelung der Kanten, wie sie zum LIFO-Prinzip des Stacks passt.

Dieser Zusammenhang zwischen Datenstrukturen und Planarität reicht sehr tief. Bernhart und Kainen haben gezeigt, dass die Graphen mit einem Stack-Layout (*Stack-Graphen*) genau die Graphen sind, die eine planare Zeichnung besitzen in der alle Knoten an der Außenfläche liegen [BK79]. In einem Doppelstack-Layout wird jede Kante in einem von zwei Stacks verarbeitet und es kann ähnlich dargestellt werden wie in Abb. 1(a), wobei die Kanten oberhalb der gepunkteten Linie im ersten und die Kanten unterhalb im zweiten Stack verarbeitet werden. Die Doppelstack-Graphen sind die Subgraphen von planaren Graphen mit einem Hamiltonkreis [BK79], d.h. einem geschlossenem Weg durch den Graphen in dem jeder Knoten genau einmal besucht wird.

In einem Queue-Layout müssen die Kanten auf einer Seite einer linear verketteten Liste eingefügt und auf der anderen Seite entfernt werden. Ein Queue-Layout ist in Abb. 1(c) dargestellt, wobei das Einfügen von Kante $\{2, 3\}$ (gestrichelt) das Queue-Layout zerstört. Das FIFO-Prinzip der Queue stellt sich in Kantenkreuzungen dar: Zwei Kanten, die keinen gemeinsamen Endpunkt besitzen, sich aber zu einem Zeitpunkt gleichzeitig in der Queue befinden, müssen sich in dieser Darstellung kreuzen. Es darf also keine vollständigen Schachtelungen geben, wie z.B. bei $\{1, 5\}$ und $\{2, 3\}$. In Abb. 1(c) ist das FIFO-Prinzip zwar durch die Kreuzungen sichtbar, diese erschweren allerdings auch die Lesbarkeit der Darstellung. Eine planare Darstellung von Queue-Graphen ist deshalb wünschenswert

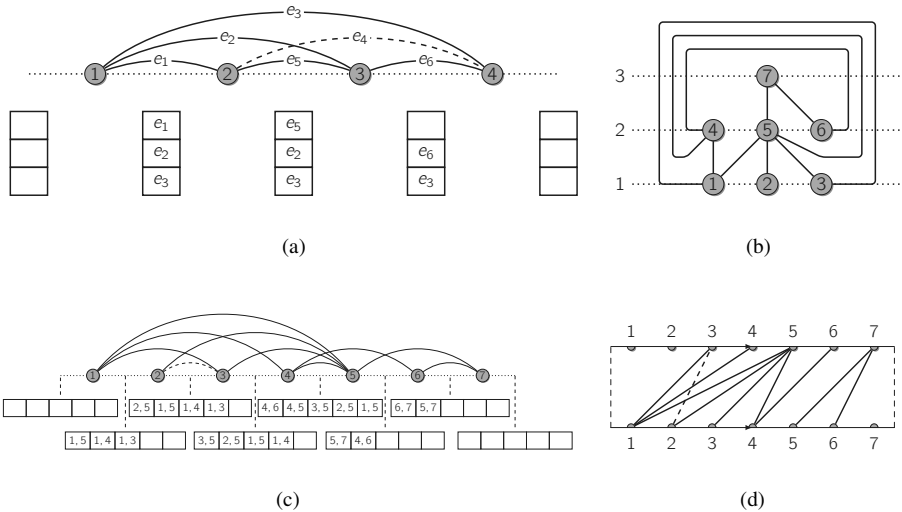


Abbildung 1: Beispiele für Graph-Layouts im Stack, Doppelstack und der Queue.

und auch möglich: Heath und Rosenberg haben in [HR92] die Queue-Graphen als die Graphen charakterisiert, die eine *Level-planare Zeichnung mit Bögen* („arched leveled-planar“) besitzen (Abb. 1(b)). In dieser Darstellung werden die Knoten auf horizontale Level platziert. Kanten sind nur zwischen benachbarten Level möglich oder als Bögen vom Knoten ganz links zu Knoten, die von rechts unterhalb desselben Levels ohne Kreuzungen erreichbar sind. Zwar ist diese Art der Darstellung planar, ein fehlerhaftes Queue-Layout ist mit ihr allerdings nur schwer zu erkennen: Die das Queue-Layout zerstörende Kante $\{2,3\}$ kann zwar nicht in legaler Weise in Abb. 1(b) eingefügt werden, aber eine andere Aufteilung der Knoten auf Level könnte prinzipiell zu einem gültigen Layout führen.

Die Dissertation erweitert das Studium von Graph-Layouts auf die Deque, einer doppelt verketteten Liste, in die Elemente auf beiden Seiten, Kopf und Fuß, eingefügt und entfernt werden können. Planarität spielt auch für die Deque eine wichtige Rolle. Um dies zu sehen wird eine neue Darstellungsart auf der Zylinderoberfläche eingeführt in der eine ungültige Deque-Operation sofort als Kreuzung sichtbar ist. Diese führt auch zu einer neuen Darstellungsart von Queue-Graphen, die die oben geschilderten Probleme nicht aufweist.

2.2 Deque-Graphen und linear-zylindrisch Zeichnungen

In einer linear-zylindrischen Zeichnung auf der Oberfläche des rollenden Zylinders werden die Knoten auf der *Frontlinie*, einem Liniensegment parallel zur Zylinderachse, auf unterschiedlichen Positionen platziert und die Kantenkurven dürfen die Frontlinie nicht schneiden. Ein Beispiel für eine linear-zylindrische Zeichnung zeigt Abb. 2(a) in der die Frontlinie gepunktet dargestellt ist. Eine ebene Darstellung erhält man, indem man die

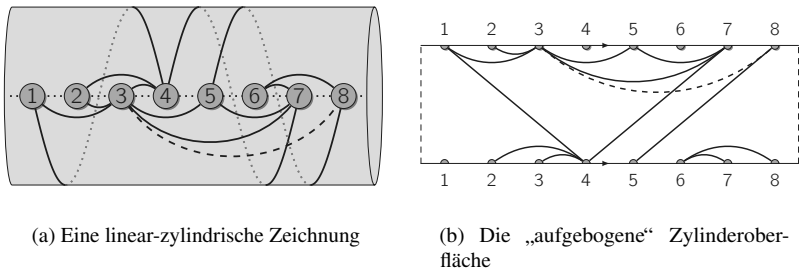


Abbildung 2: Linear-zylindrische Zeichnungen

Zylinderoberfläche entlang der Frontlinie aufschneidet und den Zylindermantel aufbiegt. Abbildung 2(b) zeigt das Resultat wobei die obere und untere Kante des Rechtecks identifiziert sind und mit der Frontlinie übereinstimmen. Graphen, die eine linear-zylindrische Zeichnung ohne Kreuzungen erlauben, heißen *linear-zylindrisch planar*. Ihre Charakterisierung bildet folgendes Theorem der Dissertation:

Theorem. *Die linear-zylindrisch planaren Graphen sind genau die Deque-Graphen.*

Dem Beweis liegt folgende Idee zugrunde. Wie in Abbildungen 2(a) und 2(b) zu sehen ist, beginnt bzw. endet jede Kantenkurve in einer linear-zylindrischen Zeichnung entweder ober- oder unterhalb der Frontlinie. Beispielsweise beginnt und endet Kante $\{2,4\}$ oberhalb während Kante $\{1,4\}$ unterhalb beginnt und oberhalb endet. Aus einer linear-zylindrisch planaren Zeichnung lässt sich ein gültiges Deque-Layout wie folgt ableiten. Die Reihenfolge der Knoten auf der Frontlinie definiert das lineare Layout, d.h. die Verarbeitungsreihenfolge der Knoten. Indem man den Bereich oberhalb der Frontlinie mit dem Kopf und der Bereich unterhalb der Frontlinie mit dem Fuß der Deque assoziiert, lässt sich für jede Kante festlegen auf welcher Seite der Deque sie eingefügt und entfernt wird. Kante $\{2,4\}$ in Abbildungen 2(a) und 2(b) wird somit am Kopf der Deque eingefügt und entfernt während Kante $\{1,4\}$ am Fuß eingefügt und vom Kopf entfernt wird.

Konsequenterweise muss sich eine in einer linear-zylindrischen Zeichnung unvermeidbare Kreuzung in einer ungültigen Deque-Operation äußern. Dazu ist folgende Beobachtung hilfreich: eine gedachte vertikale Linie, platziert zwischen den Knoten 2 und 3 in Abb. 2(b), schneidet die Kanten $\{2,4\}$, $\{1,4\}$, $\{1,3\}$ und $\{2,3\}$ in dieser Reihenfolge von unten nach oben. Dies entspricht genau dem Inhalt der Deque in dieser Reihenfolge vom Kopf zum Fuß unmittelbar vor der Verarbeitung von Knoten 3. Wird diese gedachte vertikale Linie nun unmittelbar vor die Kreuzung zwischen den Kanten $\{3,8\}$ und $\{4,7\}$ gelegt, so ergibt sich für den Inhalt der Deque $\{5,8\}$, $\{4,7\}$, $\{3,8\}$, $\{3,7\}$ und $\{5,7\}$. Nach Abb. 2(b) muss die Kante $\{4,7\}$ vor der Kante $\{3,8\}$ vom Fuß der Deque entfernt werden, da beide unterhalb der Frontlinie enden. Dies ist aber nur möglich, indem die Kanten in der Deque ihre Positionen tauschen, was keiner gültigen Deque-Operation entspricht.

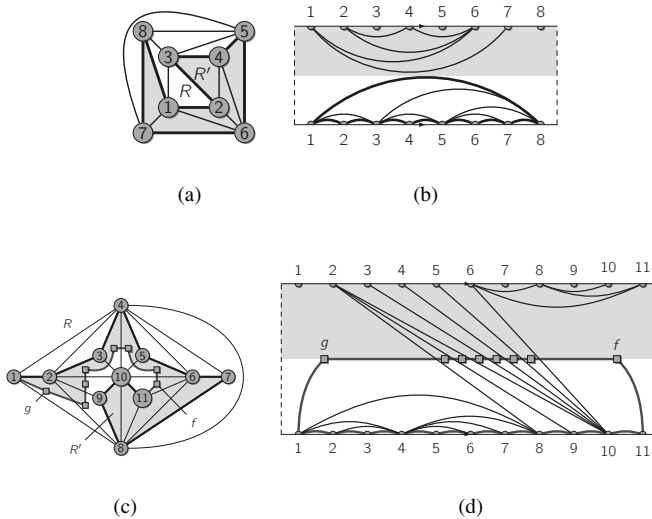


Abbildung 3: Der Unterschied zwischen Doppelstack- und Deque-Graphen im Dualgraphen

2.3 Deque-Graphen und Doppelstack-Graphen

Verlangt man bei einer Deque, dass Elemente immer nur auf der Seite entfernt werden auf der sie auch eingefügt wurden, so erhält man den Doppelstack. Ein Doppelstack-Graph ist somit auch immer ein Deque-Graph. Eine Deque erlaubt zusätzlich noch *Queue-Kanten*, die sie durchwandern. Dies wirft die Frage nach dem Unterschied zwischen Doppelstack- und Deque-Graphen auf, dessen Antwort im folgenden skizziert wird.

In Abb. 2(b) lassen sich die Kanten $\{1,2\}$, $\{4,5\}$, $\{5,6\}$ und $\{7,8\}$ einfügen ohne Kreuzungen zu erzeugen. Der daraus resultierende Graph besitzt einen Hamiltonweg entlang der Frontlinie, d.h., einen einfachen Weg, der jeden Knoten einmal besucht. Außerdem gilt mit den Ergebnissen von oben, dass jeder Deque-Graph planar ist. Somit ist jeder Deque-Graph der Subgraph eines planaren Graphen mit einem Hamiltonweg. In der Dissertation wird auch die umgekehrte Richtung gezeigt:

Theorem. *Die Deque-Graphen sind die Subgraphen von planaren Graphen mit einem Hamiltonweg.*

Da die Doppelstack-Graphen die Subgraphen von planaren Graphen mit einem *Hamiltonkreis* sind [BK79], lässt sich oben gestellte Frage beantworten: Die Möglichkeit der Deque Queue-Kanten zu verarbeiten entspricht dem Unterschied zwischen Hamiltonkreisen und -wegen in planaren Graphen.

Diese Aussage lässt sich mit Hilfe des Dualgraphen präzisieren. Abbildung 3(a) zeigt eine planare Zeichnung eines Graphen mit einem Hamiltonkreis (fett hervorgehobene

Kanten). Es handelt sich dabei also um einen Doppelstack-Graphen und somit auch um einen Deque-Graphen mit einer linear-zylindrisch planaren Zeichnung ohne Queue-Kanten (Abb. 3(b)). Die linear-zylindrisch planare Zeichnung lässt sich hierbei direkt aus Abb. 3(a) ableiten, indem man als lineares Layout die Reihenfolge der Knoten auf dem Hamiltonkreis verwendet. Welcher Knoten der erste des linearen Layouts ist, ist dabei unerheblich. Der Hamiltonkreis teilt die Ebene in Abb. 3(a) in zwei Regionen R und R' . Identifiziert man nun R mit dem Kopf und R' mit dem Fuß der Deque, so kann daraus für jede Kante abgelesen werden auf welcher Seite sie verarbeitet wird. Kante $\{1, 3\}$ befindet sich vollständig in R und wird damit am Kopf eingefügt und entfernt bzw. beginnt und endet oberhalb der Frontlinie in Abb. 3(b). Die Region der Kanten auf dem Hamiltonkreis sei auf R festgelegt.

Abbildung 3(c) zeigt den Goldner-Harary-Graphen [GH75], den kleinsten maximal planaren Graphen ohne Hamiltonkreis. Dieser Graph besitzt einen Hamiltonweg (hervorgehobene Kanten) und ist somit ein Deque-Graph (jedoch kein Doppelstack-Graph). Die entsprechende linear-zylindrisch planare Zeichnung ist in Abb. 3(d) dargestellt und sie wird wie folgt ermittelt. Die Endpunkte des Hamiltonwegs 1 und 11 können über den Dualgraphen miteinander verbunden werden. Dazu verbindet man Knoten 11 mit einem anliegenden Flächenknoten, hier f , und wandert dann auf dem Dualgraphen zu einem Flächenknoten, der an Knoten 1 anliegt, hier g . Dabei wird keine der Kanten des Hamiltonwegs geschnitten und so ergibt sich auch hier eine Teilung der Ebene in zwei Regionen R und R' . Wie zuvor wird R mit dem Kopf und R' mit dem Fuß der Deque identifiziert. So befindet sich z.B. Kante $\{1, 4\}$ vollständig innerhalb von R und ist somit eine Kante die am Kopf eingefügt und entfernt wird. Kante $\{2, 8\}$ beginnt innerhalb von R' und endet innerhalb von R . Dies entspricht in Abb. 3(d) einer Kante, die sich um den Zylinder herumbewegt, und somit einer Queue-Kante, die am Fuß eingefügt und am Kopf der Deque entfernt wird. Der Unterschied zwischen Doppelstack und Deque entspricht also dem Verbindungslinie zwischen den Endpunkten des Hamiltonwegs, der die Queue-Kanten schneidet.

2.4 Linear-zylindrische Zeichnungen von Queue-Graphen

Da die Queue ein Spezialfall der Deque ist, in der die Elemente am Kopf eingefügt und am Fuß entfernt werden, lassen sich linear-zylindrische Zeichnungen direkt auf Queue-Layouts anwenden. Abbildung 1(d) zeigt die linear-zylindrisch planare Zeichnung des Queue-Layouts aus Abbildungen 1(b) and 1(c). Letztere Darstellungsarten repräsentieren das FIFO-Prinzip bzw. dessen Verletzung nicht ausreichend (Abb. 1(b)) oder sind wegen vieler Kreuzungen unübersichtlich (Abb. 1(c)). Die linear-zylindrisch planare Darstellung in Abb. 1(d) hat diese Probleme nicht, da das FIFO-Prinzip durch die Planarität der Zeichnung unmittelbar repräsentiert ist. Somit ist aus den Kreuzungen (z.B. durch Kante $\{2, 3\}$) sofort ersichtlich, ob und durch welche Kanten das Queue-Layout zerstört wird.

2.5 Weitere Ergebnisse und Zusammenfassung

Für das Entscheidungsproblem „Ist ein Graph ein Deque-Graph?“ wird gezeigt, dass es \mathcal{NP} -vollständig ist. Das ergänzt die entsprechenden Resultate zur \mathcal{NP} -Vollständigkeit

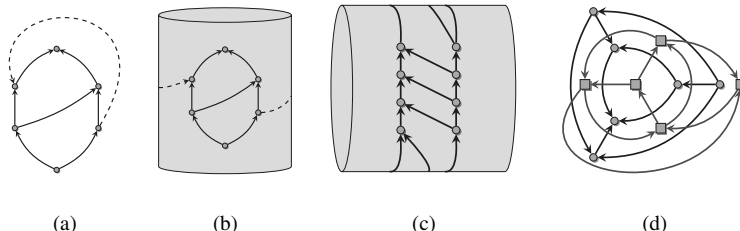


Abbildung 4: Aufwärtsplanare Zeichnungen und gerichtete Dualgraphen

bei Doppelstack- und Queue-Graphen [Wi82, HR92]. Außerdem wird in der Dissertation gezeigt, dass Queue-Graphen eine „selbst-duale“ Struktur besitzen, d.h. die Struktur von Queue-Graphen spiegelt sich in deren Dualgraphen wider.

Nicht jeder maximal planare Graph ist ein Deque-Graph [He07]. Dies führt zu der Frage nach einer Operation um die die Deque ergänzt werden muss, so dass sie genau alle planaren Graphen charakterisiert. In der Dissertation wird die Deque um eine „Split“-Operation erweitert was zur *teilbaren Deque* führt. Es wird gezeigt, dass ein Graph genau dann planar ist wenn er ein Layout in der teilbaren Deque besitzt. Für den Beweis wird ein Algorithmus entworfen, der prüft ob eine gegebene zyklische Anordnung der Kanten um jeden Knoten eine planare Zeichnung erlaubt.

In der Dissertation wurde die Klassifikation planarer Graphen mittels Datenstrukturen vervollständigt. Als nützliches Werkzeug zum Studium der Arbeitsweise fundamentaler Datenstrukturen wurden linear-zylindrische Zeichnungen eingeführt. Diese bringen nicht nur Einsichten zur Arbeitsweise der Deque sondern auch zur Queue und zum Doppelstacks.

3 Zylindrisch aufwärtsplanare Graphen

Der Standard-Algorithmus zum Zeichnen gerichteter Graphen ist das Verfahren von Sugiyama [STT81]. Eine grundlegende Idee darin ist, die Kantenrichtung in eine geometrische Richtung zu transformieren. Verlaufen alle Kantenkurven streng monoton steigend in y -Richtung von ihrem Start- zu ihrem Endpunkt so spricht man von einer *Aufwärtszeichnung* und im planaren Fall von einer *aufwärtsplanaren Zeichnung*. Letztere spielen im Graph Drawing eine wichtige Rolle. Abbildung 4(a) (ohne die gestrichelte Kante) zeigt einen **UP**-Graphen („upward planar graph“). Fügt man die gestrichelten Kante ein, so ist keine aufwärtsplanare Zeichnung in der Ebene mehr möglich [Di99, S. 171]. Auf der Oberfläche des stehenden Zylinders hingegen kann die gestrichelte Kante um den Zylinder herumgeführt werden (Abb. 4(b)). Graphen die eine solche Zeichnung zulassen heißen **SUP** („standing upward planar“). Während im ungerichteten Fall die Ebene und der Zylindermantel bezüglich Planarität äquivalent sind, ist im gerichteten Fall überraschenderweise der stehende Zylinder mächtiger als die Ebene.

Die Dissertation beschäftigt sich mit Aufwärtsplanarität auf dem rollenden Zylinder (**RUP**, „rolling upward planar“). Hier bewegen sich die Kantenkurven in einer Richtung um die Zylinderachse herum (Abb. 4(c)). Während **SUP**-Graphen azyklisch sind, können **RUP**-Graphen Zyklen enthalten. **UP**- und **SUP**-Graphen wurden in der Vergangenheit intensiv auf charakterisierende und algorithmische Eigenschaften hin untersucht. Im Gegensatz dazu rückte die Klasse der **RUP**-Graphen erst in jüngerer Zeit bei der Erweiterung des Sugiyama-Frameworks auf zyklische Darstellungen [Ba12, Br14] in den Fokus.

3.1 Das Ampèresches Gesetz aufwärtsplanarer Graphen

Für die Charakterisierung aufwärtsplanarer Graphen spielen *azyklische Dipole* eine wichtige Rolle. Ein azyklischer Dipol ist ein azyklischer Graph mit genau einer Quelle s und genau einer Senke t (Abbildungen 4(a) and 4(b)). Ein Graph ist genau dann **SUP**, wenn er durch Kanten planar zu einem azyklischen Dipol ergänzt werden kann [Ha01]. Verlangt man zusätzlich die Kante von s nach t , so erhält man **UP**, die aufwärtsplanaren Graphen in der Ebene [Ke87]. Diese Art der Charakterisierung kann für **RUP**-Graphen nicht angewandt werden, da diese Zyklen enthalten können. Azyklische Dipole spielen dafür bei den gerichteten Dualgraphen von **RUP**-Graphen eine wichtige Rolle. Abbildung 4(d) zeigt eine planare Zeichnung eines gerichteten Graphen (Knoten runder Form) und dessen Dualgraph (Knoten eckiger Form). Die Richtung einer Kante im Dualgraphen ist dabei so gewählt, dass in Blickrichtung der dualen Kante die (primale) Kante von rechts nach links verläuft. Wie in Abb. 4(d) zu sehen ist, ist der primale Graph azyklisch während der Dualgraph stark zusammenhängend ist, d.h. jedes Paar von Knoten ist durch einen gerichteten Pfad verbunden. Dies gilt allgemein: Azyklizität und starker Zusammenhang tauschen ihre Rollen vom Primal- zum Dualgraphen.

Der erste Schritt der vollständigen Charakterisierung von **RUP**-Graphen ist das Ampèresche Gesetz aufwärtsplanarer Graphen, dessen Name eine physikalische Interpretation nahelegt. Abbildung 5(a) zeigt einen elektrischen Leiter an dem eine Spannungsquelle einen Stromfluss \vec{I} erzeugt. Das Ampèresche Gesetz besagt, dass dieser Strom ein Magnetfeld \vec{B} induziert, das sich um den Leiter windet. Abbildung 5(b) zeigt einen azyklischen Dipol, der aufwärtsplanar auf dem stehenden Zylinder gezeichnet ist (Knoten runder Form). Legt man nun eine Spannungsquelle an die „elektrischen Kontakte“ s und t an, so induziert dies den gerichteten Dualgraphen (Knoten rechteckiger Form), der sich wie das Magnetfeld um den Zylinder windet. Der Dualgraph ist ein stark zusammenhängender **RUP**-Graph. Stehender und rollender Zylinder tauschen somit ihre Rollen vom Primal- zum Dualgraphen.

Theorem. *Ein stark zusammenhängender Graph ist genau dann ein **RUP**-Graph wenn sein Dualgraph ein azyklischer Dipol ist.*

Diese Charakterisierung stark zusammenhängender Graphen wird in zwei weiteren Schritten auf allgemeine Graphen erweitert. Zunächst werden abgeschlossene Graphen betrachtet, d.h. Graphen ohne Quellen und Senken. Für diese wird eine „high-level“ Beschreibung eingeführt, die einen abgeschlossenen Graphen in seine azyklischen Komponenten (Transits) und seine stark zusammenhängenden Komponenten (Verbünde) aufteilt. Es wird gezeigt,

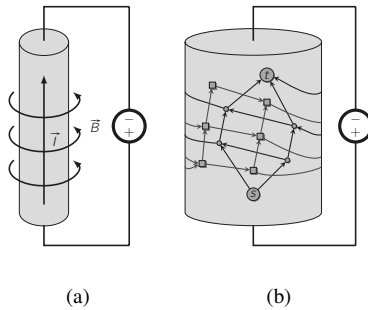


Abbildung 5: Das Ampèresche Gesetz des Elektromagnetismus und für aufwärtsplanare Graphen

dass Transits und Verbünde ihre Rollen vom Primal- zum Dualgraph tauschen. Basierend auf diesen Ideen wird die Definition des azyklischen Dipols auf (*allgemeine*) *Dipole* erweitert, die Zyklen enthalten dürfen. Dies führt zur Verallgemeinerung des Ampèreschen Gesetzes.

Theorem. *Ein abgeschlossener Graph ist **RUP** gdw. sein Dualgraph ein Dipol ist.*

Außerdem kann jeder **RUP**-Graph durch Kanten zu einem abgeschlossenen **RUP**-Graphen ergänzt werden, was zur gewünschten Charakterisierung von **RUP**-Graphen führt:

Theorem. *Ein Graph G ist genau dann ein **RUP**-Graph wenn er der Subgraph eines abgeschlossenen planaren Graphen H ist, dessen Dualgraph ein Dipol ist.*

3.2 Ein **RUP**-Erkennungsalgorithmus für abgeschlossene Graphen

Das Erkennen von **UP**- und **SUP**-Graphen ist \mathcal{NP} -vollständig [GT01, HRK98]. Im Falle von azyklischen Dipolen, ist die Erkennung jedoch effizient lösbar. Die Komplexität scheint somit in der Ergänzung des Graphen zu einem planaren azyklischen Dipol zu liegen. Für **RUP**-Graphen ist die Situation ähnlich: Im Allgemeinen ist die Erkennung \mathcal{NP} -vollständig [Br14]. Für den Fall von abgeschlossenen Graphen wird in der Arbeit ein Linearzeit-Algorithmus entwickelt, der entscheidet ob der Graph **RUP** ist. Der Algorithmus verwendet die vorher beschriebene Charakterisierung von **RUP**-Graphen über deren Dualgraphen. Dazu werden bekannte Strukturen, wie der block-cut tree und der SPQR-Baum [DBT96], so erweitert, dass sie auf gerichtete Dualgraphen anwendbar sind.

3.3 Zusammenfassung

Die gewonnenen Erkenntnisse und entwickelten algorithmischen Werkzeuge sind für das Studium von Aufwärtsplanarität über Zylinderoberflächen hinaus nützlich. So lässt sich der Erkennungsalgorithmus leicht für die Aufwärtsplanarität auf dem Torus erweitern. Außerdem können die Erweiterungen von block-cut trees und SPQR-Bäume auf gerichtete Dualgraphen durch ihre Allgemeinheit auch in anderen Kontexten Anwendung finden.

Literatur

- [Au14] Auer, Christopher: Planar Graphs and their Duals on Cylinder Surfaces. Dissertation, Lehrstuhl für Informatik mit Schwerpunkt Theoretische Informatik, Universität Passau, Germany, Januar 2014.
- [Ba12] Bachmaier, Christian; Brandenburg, Franz J.; Brunner, Wolfgang; Fülöp, Raymund: Drawing Recurrent Hierarchies. *J. Graph Alg. App.*, 16(2):151–198, 2012.
- [BK79] Bernhart, Frank; Kainen, Paul Chester: The Book Thickness of a Graph. *J. Combin. Theory, Ser. B*, 27(3):320–331, 1979.
- [Br14] Brandenburg, Franz J.: Upward Planar Drawings on the Standing and the Rolling Cylinders. *Computational Geometry*, 47(1):25–41, 2014.
- [DBT96] Di Battista, Giuseppe; Tamassia, Roberto: On-Line Planarity Testing. *SIAM J. Comput.*, 25(5):956–997, 1996.
- [Di99] Di Battista, Giuseppe; Eades, Peter; Tamassia, Roberto; Tollis, Ioannis G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
- [GH75] Goldner, A.; Harary, Frank: Note on a Smallest Nonhamiltonian Maximal Planar Graph. *Bull. Malaysian Math. Soc.*, 6(1):41–42, 1975.
- [GT01] Garg, Ashim; Tamassia, Roberto: On the Computational Complexity of Upward and Rectilinear Planarity Testing. *SIAM J. Comput.*, 31(2):601–625, 2001.
- [Ha01] Hashemi, S. Mehdi: Digraph Embedding. *Discrete Math.*, 233(1–3):321–328, 2001.
- [He07] Helden, Guido: Hamiltonicity of Maximal Planar Graphs and Planar Triangulations. Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, 2007.
- [HR92] Heath, Lenwood Scott; Rosenberg, Arnold L.: Laying Out Graphs Using Queues. *SIAM J. Comput.*, 21(5):927–958, 1992.
- [HRK98] Hashemi, S. Medi; Rival, Ivan; Kisielewicz, Andrzej: The Complexity of Upward Drawings on Spheres. *Order*, 14:327–363, 1998.
- [Ke87] Kelly, David: Fundamentals of Planar Ordered Sets. *Discrete Math.*, 63:197–216, 1987.
- [STT81] Sugiyama, Kozo; Tagawa, Shojiro; Toda, Mitsuhiro: Methods for Visual Understanding of Hierarchical System Structures. *IEEE Trans. Syst., Man, Cybern.*, 11(2):109–125, 1981.
- [Wi82] Wigderson, Avi: The complexity of the Hamiltonian Circuit Problem for Maximal Planar Graphs. Bericht, Department of EECS, Princeton University, 1982.



Christopher Auer wurde am 6. April 1982 in Passau geboren. Er studierte zunächst an der Ostbayerischen Technischen Hochschule Regensburg (Dipl.-Inf.) und der Universität Passau (MSc.) Informatik und schloß beides mit Auszeichnung ab. Vom Wintersemester 2008 an bis zum Wintersemester 2014 forschte Herr Auer am Lehrstuhl von Prof. Dr. Franz J. Brandenburg im Bereich der Graphentheorie und insbesondere im Forschungsgebiet „Zeichnen von Graphen“, zu dem er mehrere Fachartikel auf internationalen Konferenzen vorstellte. Seine Promotion schloss er im November 2014 ab. Seit Juli 2014 arbeitet Herr Auer bei Ravensburger Digital an der Entwicklung von hybriden Spielkonzepten.