# Planare Graphen - Planare Einbettung

Manuel Frohn

RWTH Aachen University, Aachen, Germany
manuel.frohn@rwth-aachen.de

**Zusammenfassung.** In dieser Ausarbeitung behandeln wir Planare Graphen. Diese zeichnen sich dadurch aus, dass sie sich in die Ebene einbetten lassen, ohne dass sich Kanten kreuzen. Im weiteren wird auch ein $\mathcal{O}(n)$ Laufzeit Algorithmus zur Errechnung einer solchen Einbettung vorgestellt.

**Schlüsselwörter:** Planare Graphen · Planare Einbettung

## 1 Motivation

Die Klasse der Planaren Graphen ist sowol für die innermathematische Untersuchung von Graphen, als auch zur Modelierung praktischer Netzwerke, wie zum Beispiel Straßennetze,Platinenschaltkreise oder Landkarten wichtig. So dient die Planarität von Graphen als Bedingung für die Funktionalität und/oder Effizienz einiger Algorithmen. Zu diesen zählt zum Beispiel der Algorithmus zur Berechnung einer 4-Färbung in polinomieller Zeit.

Die Frage nach einer Planaren Einbettung ist nun eine natürliche Frage. Sie ist zum Beispiel notwendig um zum Beispiel ein kurzschlussfreien Platinenschaltkreise zu erstellen.

## 2 Einführung

**Definition 1 (Planarität).** *Ein Graph G heißt planar, wenn man den Graphen so auf eine Ebene zeichnen kann, dass sich seine Kanten nicht schneiden.*

Planare Graphen sind unter anderem durch die sehr graphische Natur ihrere Definition, eine Graphklasse, die schon sehr lange untersucht wird. Um ein besseres Verständnis für die Graphklasse zu entwickeln wollen wir uns nun die Abbildungen aus Fig. 1 betrachten

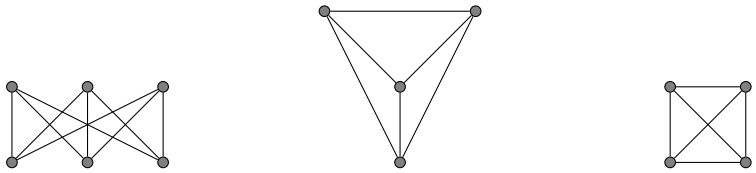


Abb. 1: Der $K_{3,3}$ (a) ist nicht planar, der $K_4$ (b) jedoch schon, lässt sich aber trozdem mit Kantenüberschneidung zeichnen (c)

(a) zeigt den sogenanten $K_{3,3}$, den volständigen bipariten Graphen, mit jeweils 3 Knoten pro Partition. Dieser ist nicht Planar. Egal wie man den Graphen zeichnet, man wird nicht in der Lage sein ihn so zu zeichnen, dass sich keine seiner Kanten schneiden. Betrachten wir nun (b). Wir sehen den $K_4$, den volständigen Graphen, mit 4 Knoten. Dieser ist Planar, was sich in der Abbildung auch sehr leicht erkennen lässt. Man möchte nun vileicht glauben, dass es sich bei der Planarität um eine Trivial prüfbare Eigenschaft handelt. Jedoch, wie wir in (c) erkennen können, lässt sich nicht aus jeder Zeichnung eines Planaren Graphen direkt dessen Planarität erkennen. Bevor wir uns aber damit beschäftigen, wie man, auch algorithmisch, feststellt, ob ein Graph Planar ist, bedarf es einem kleinem Exkurs.

## 3 Minore

**Definition 2 (Teilgraph).** *Ein Graph $G' = (V', E')$ heißt Teilgraph von $G = (V, E)$, wenn gilt:*

$$V' \subset V \wedge \forall e = (a, b) \in E \text{ mit } a, b \in V' \text{ gilt } e \in E'$$

Ein Teilgraph $G'$ ist also ein Graph, der aus einem Teil der Knoten von G unduziert wird.

**Definition 3 (Kantenkontraktion).** *Gegeben ein Graph $G = (V, E)$ und eine Kante $e = v, w \in E$, ist das Resultat der Kontraktion von e, der Graph $G' = (V', E')$ mit $V' = (V \setminus e) \cup \{n\}$ und $E' = (E \setminus \{v \in e \vee w \in e \mid e \in E\}) \cup \{\{n, z\} \mid z, v \in e \in E \vee z, w \in e \in E\}$*

Die Kontraktions-Operation soll im weiteren betrachtet werden. Schauen wir uns hierzu Abbildung 2 an.

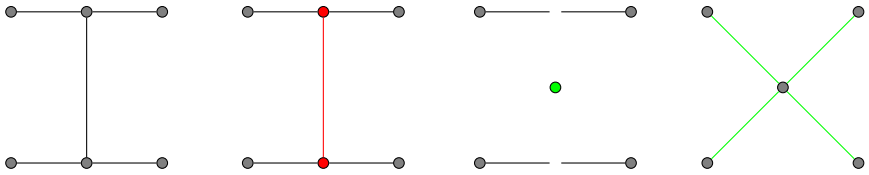

Abb. 2: Kantenkontaktion an einem Beispiel

Hier sehen wir wie einen Kantenkontraktion funktioniert. Und zwar verschmiltz man die beiden, durch die Kante verbundenen Knoten zu Einem. Dazu enfernt man zuerst die zu kontrahierende Kante, und ersetzt die beiden Knoten die sie verbindet, durch einen neuen Knoten. Danach ersetzt man jede Kante, die zufor mit einem der beiden verschmolzenen Knoten verbunden war, durch eine neue, die nun statdessen mit dem neuen Knoten verbunden ist. Betrachten wir nun das Konzept des Minors:

**Definition 4 (Minor).** *M heißt Minor von G wenn M aus einem Teilgarphen von G, durch Kantenkontaktion hervorgeht.*

Ein Minor ist nun also das Resultat einer Reduktion eines Teilgraphen. So ist zum Beispiel der Dreiecksgraph (a) aus Figur 3 ein Minor des Graphen (b)

(a) (b)

Abb. 3: Minor Beispiel

## 4 Wichtige Sätze

Für Planare Graphen gibt es zwei Wichtige Sätze, die wir betrachten wollen.

**Theorem 1 (Eulerscher Polyedersatz).**
*Für alle zusammenhängende Planare Graphen mit $|V|$ Knoten, $|E|$ Kanten und $|F|$ Flächen gilt:*

$$|V| - |E| + |F| = 2 \Leftrightarrow |E| \leq 3|V| - 6 \wedge |F| \leq 2|V| - 4$$

Beachte das mit Flächen, die von Kanten eingeschlossennen Flächen, plus die sogenanten außere Fläsche gemeint ist. Wie viele es von diesen gibt ist keineswegs einfach zu berechnen, daher wird meisst nur der erste Teil der Umformung ($|E| \leq 3|V| - 6$), als notwendige Bediengung für Planarität, verwendet.

**Theorem 2 (Satz von Kuratowski).**
*Ein Graph ist genau dan Planar, wenn er den $K_{3,3}$ und den $K_5$ nicht als Minore enthält.*

Der Satz von Kuratowski liefert anders als der Eulerscher Polyedersatz eine tatsächliche Grundlage zu Ideentiefizierung von nicht planaren Graphen. Weiter dient er einiegen Algorithmen, so auch dem der im weiteren Vorgestellt wird als Teil des Richtigkeitsbeweises

(a) $K_{3,3}$ (b) $K_5$

Abb. 4: Kuratowski Minore

## 5 Der Einbettungsalgorithmus

Der im folgenden vorgestellte Algorithmus, von Myrvold und Boyer, ist in der Lage in $\mathcal{O}(n)$ Laufzeit eine Planare Einbettung des Eingabegraphen zu errechnen. Der Algorithmus bettet hierzu Kante für Kante, im äußeren Gebiet ein, bis entweder der gesamte Graph eingebettet ist, oder der teils eingebettete Graph ein Kuratowski Minor enthält, in diesem Fall scheitert der Algorithmus und bricht ab.

### 5.1 Bikomponenten und Seperatoren

Ein wichtiger Teil des Algorithmus sind Bikomponenten. Eine Bikomponente ist ein zusammenhängender maximaler Teilgraph G' von G, ohne Separatoren. Ein Separator ist ein Knoten, der bei Entfernung, dazu führen würde, dass der Graph nicht mehr zusammenhängend ist. Bikomponenten sind wichtig, da gilt für planare Graphen gilt:

**Lemma 1.** *Ein Graph ist genau dann planar, wenn seine Bikomponenten planar sind*

**Definition 5 (Bikomponente).** *Ein maximaler Teilgraph G' von G ist einen heißt Bikomponente von G, wenn G' keine Separatoren enthält*

**Definition 6 (Separator).** *Gegeben ein Graph $G = (V, E)$ heißt ein Knoten $u \in V$ Separator, wenn für alle Pfade $p = v \overset{*}{\Rightarrow} w$ mit $v, w \in V$ gilt: $u \in p$*

Dieses Lemma bedeutet, dass der Algorithmus, beim Einbetten einer Kante, nicht den gesamten Graphen G bezüglich potenziellen Kantenüberschneidungen betrachten muss, sondern nur die Bikomponenten, die durch das Hinzufügen, der Kante verändert werden.

### 5.2 Preprocessing

Ein wichtiger Schritt, der vor dem Algorithmus, durchgeführt werden muss, ist die Berechnung eines Palmtrees von G. Hierzu berechnen wir zuerst per DepthFirstSearch Den Spanbaum von G und fügen danach die Kanten die nicht wärend des DFS durchlaufen wurden, als BackEdges in den PalmTree ein.

### 5.3 Der Algortihmus

Wir können nun damit beginnen den Graphen einzubetten. Wir beginnen, indem wir die Kanten (v,w) des Spannbaums jeweils als eigene Bikomponente einbetten. Dabei heißt die Bikomponente aus (u,v) besteht, wobei u der Spannbaumelternknoten von v ist, Elternbikomponente der Bikomponente die aus der Kante (v,w) besteht. Andersherum heißt die Bikomponente aus (v,w) Kindbikomponente, der Bikomponente aus (u,v). Um nun die Kanten hienzuzufügen, betrachten wir die Knoten in inverser DFI Ordnung und versuchen Kanten (a,b) die den Betrachteten Knoten b mit einem DFS Nochkommen a verbinden, einzubetten. Dabei müssen wir jedoch externaly activ Knoten beachten.

**Definition 7 (Externaly Activ).** *Ein Knoten w aus Bikomponente B, der DFS Nachkomme des zur Zeit bearbeiteten Knotens v ist heißt externaly activ, wenn es einen Pfad aus einer back edge und potentiel mehreren Kanten, die nicht in B sind gibt, der zu einem Vorfahren von v führt.*
*Eine Bikomponente heißt externaly activ, wenn sie einen Knoten enthält, der externaly activ ist*

Externaly Aktive sind also all jene Knoten, die später noch in einer Einbettung involviert sein werden. Folglich, nach der Idee des Algorithmus, dürfen wir Kanten nur dan hinzufügen, wenn dadurch kein externaly activ Knoten in eine innere Fläche geraten. Dies ereicht man, indem man, liegt z.B. ein externaly activ Knoten auf der rechten Seite der äußeren Fläche, die Kante in die linke Seite einbettet.

Da wir nun wissen, bei welchen Knoten wir mit vorsicht vorgehen müssen, können wir uns nun den restlichen Teil des Algorithmus angucken. Wir fahren fort, indem wir für jede hinzuzufügende Kante (a,b) von a ausgehend einen Pfad, in dem teils eingebtteten Graphen nach b suchen. Hierbei ist es möglich aus der Wurzel eine Bikomponente (dem Knoten, mit dem geringsten DFI) in die zugehörige Elternbikomponente zu treten. Passiert dies, heißt die Bikomponente aus der man kommt Relevant und wird daher im späteren betrachtet. Entlang des berechneten Pfades werden wir die Kante später einbetten. Die Knoten und Kanten diese Pfades, bis auf maximal die Endpunkte, liegen also nach dem Einbetten der Kante in einer inneren Fläche. Daher ergeben sich einige Punkte die es beim suchen eines Pfades zu beachten gilt:

1. Externaly activ Knoten dürfen nicht traversiert werden
2. Wird eine Wurzel aus zwei externaly activ Bikomponenten, oder aus zwei Richtungen beschritten,müssen zwei Pfade über die außenliegenden Knoten gefunden werden
3. Eine externaly activ Wurzel, die nicht durch die Bikomp aus der man kommt externaly activ ist, darf nur von einer Richtung aus beschritten werden

Nachdem wir dies für alle hinzuzufügenden Kanten gemacht haben, betrachten wir nun den Graphen, der durch die berechneten Pfade induziert wird. Auf diesen wenden wir nun wieder DepthFirstSearch an, bei dem betrachteten Knoten beginnend, mit folgendem Zusatz:

1. Füge zuerst die Kante zu einbettung hinzu, die an dem Knoten anliegt, indem du gerade bist
2. Gibt es eine relevante Bikomponte,ohne externaly activ Knoten von der der momentane Knoten die Wurzel ist, schreite in diese Bikomponente hinein
3. Gibt es eine relevante Bikomponte,mit außen externaly activ von der der momentane Knoten die Wurzel ist, schreite in diese Bikomponente hinein