

Julius-Maximilians-Universität Würzburg
Institut für Informatik
Lehrstuhl für Informatik I
Effiziente Algorithmen und wissensbasierte Systeme

Bachelorarbeit

Rechtecksdarstellung gewichteter außenplanarer Graphen

Lorenz Reinhart

Eingereicht am 2. September 2014

Betreuer:
Prof. Dr. Alexander Wolff
Dipl.-Inf. Philipp Kindermann

Inhaltsverzeichnis

1. Einleitung	3
2. Maximal Außenplanare Graphen	8
2.1. Definition	8
2.2. Zeichnen der Rechtecksdarstellung	9
2.3. Beweis der Korrektheit für maximal außenplanare Graphen	12
2.4. Implementierung in Java	14
3. Außenplanare Graphen	16
3.1. Erweiterung des Ausgangsgraphen	16
3.2. Schematischer Ablauf der Zeichnung	17
3.2.1. Einfügen von Dummy-Knoten	17
3.2.2. Durchlaufen des Graphen	18
3.3. Beweis der Korrektheit für außenplanare Graphen	21
4. K-außenplanare Graphen	24
5. Abschließende Worte und Ausblick	25
A. Dateiliste der beigelegten CD	29
B. Eidesstattliche Erklärung	30

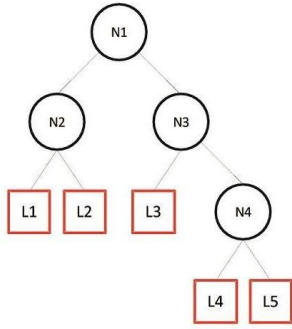
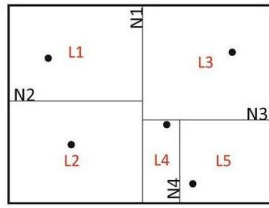
1. Einleitung

Diese Arbeit beschäftigt sich mit der Darstellung eines vorgegebenen Graphen unter Verwendung von Rechtecken. Es gibt bereits viele ähnliche Aufgaben, wie die Verteilung von Räumen innerhalb eines Grundrisses, welche zu den Packungsproblemen zählt. Dabei sollen einzelne Elemente innerhalb eines vorgegebenen Raums nicht überlappend und möglichst platzsparend untergebracht werden. Solche Probleme sind in den verschiedensten Bereichen zu finden, angefangen beim Entwurf von Stundenplänen über das effiziente Packen unterschiedlich großer Garnrollen in einer Kiste bis hin zur Aufteilung von Computerplatinen.

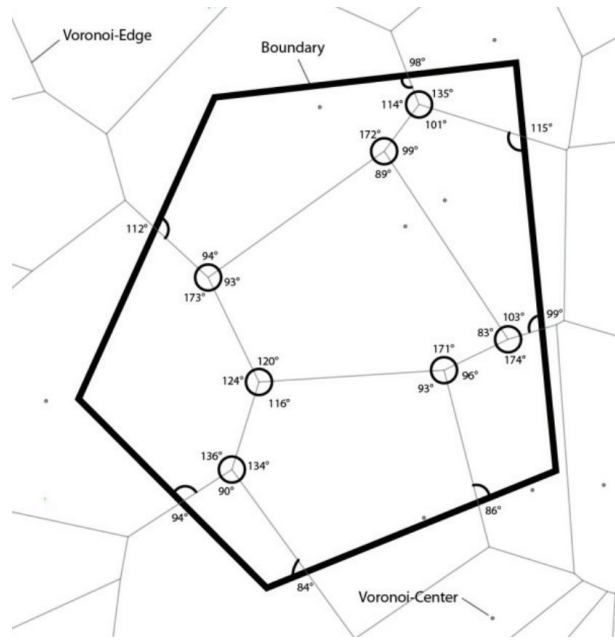
Bei der Lösung von Grundrissproblemen wird die Verteilung von Räumen in einem vorgegebenen Gebäude behandelt. In Bezug auf die automatische Berechnung solcher Probleme spricht man vom Computerbasierten Layoutentwurf bzw. dem *Computer Aided Layout Design (CALD)*. Dies ist vor allem bei komplexen Entwurfsaufgaben hilfreich, da ein breites Spektrum an Lösungen erzeugt wird, an dem sich der Architekt orientieren kann. Kleinere Teilbereiche können zudem ganz dem Computer überlassen werden. CALD-Methoden ersetzen die gestalterischen Absichten eines Architekten nicht, denn es werden „in aller Regel funktionale Kriterien verwendet, um ein Layout zu erzeugen [...]. Hierbei ist zu bemerken, dass Entwurfsaufgaben nicht allein durch Optimierungsmethoden gelöst werden können, da sich gestalterische Absichten nicht oder nur teilweise operationalisieren lassen“ [KSK12, S.17].

Es existieren bereits verschiedene Methoden des CALD, wie die Generierung von Grundrissen mittels *Kd-Bäumen* oder *Voronoi-Diagrammen*. Bei ersterem stellen Katja Knecht und Reinhard König [KSK12, Kap. 6] ein Verfahren vor, bei dem eine endliche Punktmenge P gegeben ist, wobei jeder Punkt $p \in P$ einem Raum in der zweidimensionalen Ebene entspricht. Die Unterteilung wird durch zu den Koordinatenachsen senkrechten Linien realisiert, indem der erste *Schnittlinienwert* $N1$ mittels Median oder Mittelwert von P berechnet wird. Dieser teilt P in die Punkt Mengen $P1$ und $P2$ auf, wobei die Punkte mit Koordinaten kleiner als der Schnittlinienwert $P1$ und die restlichen $P2$ zugeordnet werden. Die Kindknoten $N2$ und $N3$ des Elternknotens $N1$ stellen dabei weitere Unterteilungen der Punkt Mengen $P1$ und $P2$ dar. Diese Teilräume werden nach dem gleichen Prinzip unterteilt, bis jede Menge nur noch einen Punkt enthält. Im Beispiel 1.1a wird der Raum abwechselnd in x- und y-Richtung unterteilt, wobei hier auch andere Abfolgen möglich sind. Für jeden Schnitt N wird der Raum durch eine Wand unterteilt. Nach der Erstellung des kd-Baums müssen die Lagebeziehungen und Größen der einzelnen Räume durch Nachbearbeitung an die Vorgaben angepasst werden.

Da nicht alle Grundrisse von Gebäuden rechteckig sind, beschreiben Sven Schneider und Reinhard König [KSK12, Kap. 8], wie man eine ansprechende Raumunterteilung mit Hilfe von *Voronoi-Diagrammen* erzeugt. Anfänglich ist wieder eine Punktmenge P



(a) Mittels Kd-Baum [KSK12, Kap. 6.1]



(b) Mittels Voronoi-Diagramm [KSK12, Kap. 8.3]

Abb. 1.1.: Verschiedene Arten der Raumaufteilung

gegeben, die so genannten *Voronoi-Zentren*. Ein Punkt $p \in P$ stellt das Zentrum eines Raums im \mathbb{R}^2 dar. Alle Punkte, die sich innerhalb einer Facette befinden, sind näher an ihrem Zentrum, als an allen anderen. Die Kanten oder Grenzen einer Region sind gleich weit von zwei Zentren entfernt; siehe Abbildung 1.1b. Eine dem Optimum angenäherte Lösung findet man, indem ein Spektrum verschiedener Lösungsmöglichkeiten erzeugt und bewertet wird. Die hierfür benötigte Fitnessfunktion, welche die Summe der Differenz der einzelnen Winkel zum 90° Winkel beschreibt, wird zu diesem Zweck minimiert. Der Algorithmus bricht ab, sobald die Fitnessfunktion eine vom Nutzer vorgegebene Schranke unterschreitet. Um die topologischen Beziehungen und Raumgrößen zu berücksichtigen, kann der Nutzer einbezogen oder die Bewertungsfunktion erweitert werden.

Zudem gibt es die Möglichkeit der Grundrissberechnung mithilfe *dichter Packung*. Dabei geht es um die überschneidungsfreie Anordnung einer vorgegebene Anzahl räumlicher Elemente innerhalb eines Raumes mit minimalen Lücken. Dies kann erreicht werden, indem die Objekte eine Kollision erkennen und sich in diesem Fall voneinander abstoßen, als wären sie durch stark zusammengedrückte Federn miteinander verbunden. Sollen zwei Räume nebeneinander liegen, so ziehen sich diese an; siehe Abbildung 1.2.

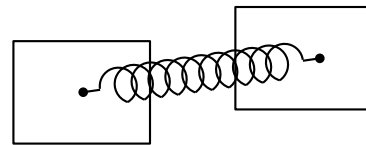


Abb. 1.2.: System virtueller Federn

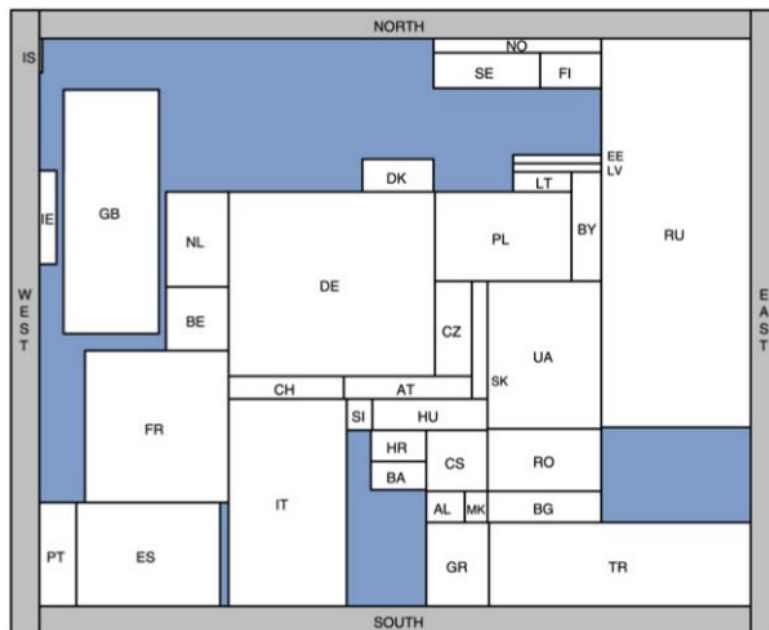


Abb. 1.3.: Die Bevölkerung Europas als Rechteck-Kartogramm [vKS07]

Ein ähnliches Vorgehen ist bei der Zeichnung von Graphen zu finden. Fruchterman und Reingold [FR91] stellen ein Verfahren vor, das sich an das Atommodell anlehnt. Knoten, die durch eine Kante miteinander verbunden sind, sollen nahe beieinander liegen, jedoch nicht zu nahe. Ähnlich verhalten sich die Nukleonen im Atomkern. Bei einer Entfernung von 1 fm (Femtometer) ist die starke Kernkraft anziehend und circa zehn mal so stark wie die elektrische Abstoßung zweier Protonen. Kommen sich die Nukleonen zu nahe, wirkt die starke Kernkraft ab ungefähr 0.4 fm abstoßend. Darauf aufbauend sollen sich alle benachbarten Knoten anziehen und zugleich alle Knoten untereinander abstoßen, um einen gleich verteilten und ansprechenden Graphen zu erhalten.

Anders gehen van Kreveld et al. [vKS07] bei der Berechnung von sogenannten *Rechteck-Kartogrammen* vor. Dabei geht es um die Darstellung von Karten, wobei jede Region durch ein Rechteck repräsentiert wird. Die Fläche der einzelnen Rechtecke bildet dabei eine geographische Größe ab, wie zum Beispiel die Anzahl an Einwohnern pro Land oder die Länge aller Autobahnen einer Region. Es ist möglich, dass durch die abgebildete Größe nicht alle Nachbarschaftsbeziehungen korrekt dargestellt werden können, wodurch das Verständnis der Karte gestört wird. Um eine ansprechende und möglichst korrekte Wiedergabe der darzustellenden Werte zu erreichen, wird die Darstellung anhand von Qualitätskriterien bewertet. Berücksichtigt werden der durchschnittliche sowie maximale kartographische Fehler, welcher die Differenz der momentanen Fläche des Landes auf der Karte zur angestrebten Landesgröße widerspiegelt, die Anzahl richtiger Nachbarschaftsbeziehungen, das maximale Seitenverhältnis, sowie die korrekte Lagebeziehung der einzelnen Länder zueinander. Um die Qualität der Karte zu maximieren werden die einzelnen Segmente so geändert, dass ein Land an einer Seite vergrößert und somit ein

anderes verkleinert wird, sodass sich der Fehler dieser beiden verringert. Dabei iteriert man mehrmals über alle Länder, bis eine zuvor definierte Qualität erreicht wird. Das Ergebnis ist ein Rechteck-Kartogramm, wie es in Abbildung 1.3 zu sehen ist.

In einer weiteren Arbeit zeigen Speckmann et al. [SvKF06], wie im letzten Schritt eine bessere Qualität erreicht werden kann, mithilfe linearer Programmierung abwechselnd über die x - und y -Achse. Dazu werden die einzelnen Rechtecke durch die sie begrenzenden Koordinaten beschrieben und darauf die Qualitätskriterien angewendet. Den kartographischen Fehler kann man beispielsweise über die Funktion $((x_4 - x_2) \cdot (y_4 - y_3) - A_s(R)) / (A_s(R))$ ermitteln, wobei x_2, x_4, y_3 und y_4 die Seiten des Rechtecks R und $A_s(R)$ die Zielgröße der Fläche angeben, wie in Abbildung 1.4 zu sehen. Um auf diese Funktion lineare Programmierung anwenden zu können, muss sie zuerst umgeformt und die x - oder y -Koordinate abwechselnd konstant gehalten werden. Wendet man dieses Verfahren auf alle Rechtecke an und iteriert über die Koordinatenachsen, so erhält man ein qualitativ hochwertiges Ergebnis. Im Unterschied zur Grundrissberechnung ist die topologische Verteilung der Rechtecke bei Kartogrammen schon vorher bekannt, was die Erstellung der Ausgangsdarstellung erheblich vereinfacht.

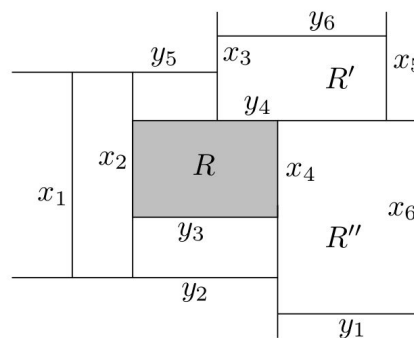


Abb. 1.4.: Lineare Begrenzungen des Rechtecks R [SvKF06]

Die vorliegende Arbeit beschäftigt sich ebenfalls mit einer Art der Grundrissprobleme, nur dass hier auf die Umrandung der Flächen verzichtet wird. Der Schwerpunkt liegt darin, die Nachbarschaftsbeziehung der Räume zueinander sowie ihre Fläche korrekt darzustellen. Eine Nachbearbeitung und die häufig damit verbundene Verwendung *evolutionärer Algorithmen*, wie sie auch beim CALD zum Einsatz kommen, soll vermieden werden.

Um dieses Ziel zu realisieren beschränken wir uns zu Beginn auf die spezielle Klasse der *maximal außenplanaren Graphen*, welche einen Teil der *außenplanaren Graphen* bildet und in Abschnitt 2.1 genauer definiert wird. Diese Graphen besitzen eine äußere Facette, an die alle Knoten angrenzen. Die inneren Facetten sind jeweils von drei Kanten umgeben. Algorithmus 1 mit der Bezeichnung `MAXOUTERPLANARRECTANGULAR(G)` in Abschnitt 2.2 durchläuft einen Suchbaum mittels Breitensuche, wobei eine zur Außenfacette adjazente Dreiecksfacette als Start gewählt wird. Die Facette wird durch drei Knoten begrenzt. Diese repräsentieren die Rechtecke, welche entsprechend des Algorithmus gesetzt werden. Anschließend besucht man die beiden Nachbarfacetten und setzt jeweils ein neues Rechteck, bis alle platziert wurden. Anders als bei den zuvor erwähnten Verfahren werden die Rechtecke von Beginn an in der vorgegebenen Größe gezeichnet. Um dies zu erreichen hält man bei jeweils zwei Rechtecken, an die noch ein Weiteres angefügt werden könnte, etwas Platz frei. Wird ein Rechteck zur Zeichnung hinzugefügt, so ist die Länge einer Seite durch die Konstruktion gegeben, wobei sich die der anderen aus der Fläche ergibt. Ein Nachteil dieses Vorgehens ist, dass ein vorgegebenes Seitenver-

hältnis der einzelnen Rechtecke nicht berücksichtigt werden kann. Im darauf folgenden Abschnitt 2.3 wird die Korrektheit des vorgestellten Algorithmus mittels Induktion über die Anzahl der gesetzten Rechtecke bewiesen. Anschließend wird anhand eines Beispiels die Implementierung unter Verwendung von Java in Abschnitt 2.4 vorgestellt.

Die Beschränkung auf die Maximalität der außenplanaren Graphen wird in Kapitel 3 aufgehoben. Zu Beginn werden die beiden Graphenklassen, der maximal außenplanaren und der außenplanaren, sowie deren Rechtecksdarstellung in Abschnitt 3.1 gegenüber gestellt. Daraus resultiert, wie in Abschnitt 3.2 beschrieben, dass der Ausgangsgraph mit Knoten erweitert werden muss, um alle Gebiete des Graphen zu erreichen. Dies geschieht in Unterabschnitt 3.2.1, in dem äußere und innere Dummy-Knoten eingeführt und zum Graphen hinzugefügt werden. Der daraus entstehende Graph wird, aufbauend auf den in Abschnitt 2.2 vorgestellten Algorithmus, nach einem bestimmten Schema durchlaufen, welches in Unterabschnitt 3.2.2 beschrieben wird. Dazu wird beim Betreten einer Dreiecksfacette zwischen normalen Knoten, sowie äußeren und inneren Dummy-Knoten unterschieden. Abschließend wird in Abschnitt 3.3 die Korrektheit des Verfahrens induktiv über die Anzahl der gesetzten Rechtecke bewiesen.

Bei der zuletzt betrachteten Graphenklasse in Kapitel 4 handelt es sich um k -außenplanare Graphen. Diese sind $k-1$ -außenplanar, wenn alle zur Außenfacette angrenzenden Knoten entfernt werden. Dabei stellt sich heraus, dass schon für die 2-außenplanaren Graphen solche existieren, die keine Rechtecksdarstellung besitzen.

Am Ende werden die Ergebnisse der Arbeit in Kapitel 5 zusammengefasst und ein Ausblick über weitere Anpassungen bei der Umsetzung der Rechtecksdarstellung geliefert.

2. Maximal Außenplanare Graphen

2.1. Definition

Ein ungerichteter *Graph* $G = (V, E)$ besteht aus der nichtleeren Knotenmenge V und der Kantenmenge E mit $V \cap E = \emptyset$. Jeder Endpunkt einer Kante $e \in E$ wird dabei genau einem Knoten zugeordnet.

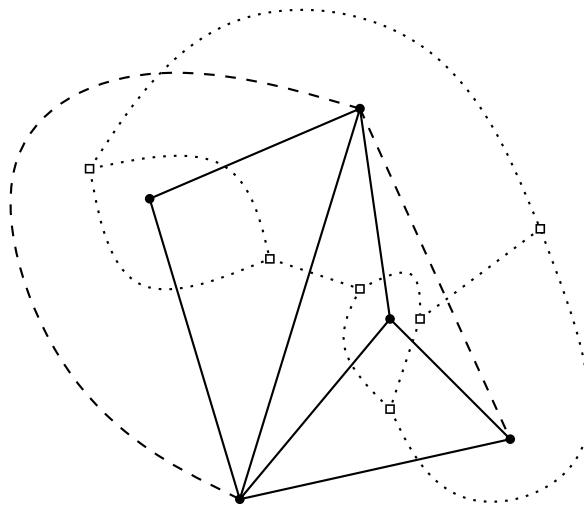


Abb. 2.1.: Ein maximal außenplanarer Graph (durchgezogene Kanten), maximal planarer Graph (gestrichelte und durchgezogene Kanten) und sein Dualgraph (gepunktete Kanten)

Die *planare Einbettung* eines Graphen im \mathbb{R}^2 ordnet jedem Knoten aus V einen Punkt in der Ebene zu. Dabei ist jede Kante ein Polygonzug im \mathbb{R}^2 und die beiden Endpunkte sind Punkte aus V . Das Innere jeder Kante soll weder eine Ecke, noch einen Punkt einer anderen Kante enthalten. Die Darstellung ist somit überschneidungsfrei.

Ein Graph $G_1 = (V_1, E_1)$ ist zu einem Graphen $G_2 = (V_2, E_2)$ *dual*, wenn die Funktion $f : V_1 \rightarrow F_2$ existiert und somit die Abbildung der Knoten V_1 auf die Facetten F_2 von G_2 , sodass jeder Knoten genau einer Facette zugeordnet wird. Außerdem werden die Kanten E_1 auf E_2 abgebildet und jede Kante schneidet sich dabei in genau einem Punkt mit der Kante, auf die sie abgebildet wird [KN12, Kap. 12.5].

Die Klasse der *planaren Graphen* sind solche, die eine planare Einbettung besitzen [KN12, Kap. 12.1]. Ein Graph heißt *außenplanar*, wenn er sich planar so in die Ebene einbetten lässt, dass alle seine Knoten am Rand ein und desselben Gebiets liegen. Aufbauend auf diesen Bezeichnungen ist ein *maximal außenplanaren Graphen* als solcher

definiert, dem keine Kante hinzugefügt werden kann, ohne Einfachheit oder Außenplanarität zu verletzen, so wie es in Abb. 2.1 dargestellt ist. Aus der Maximalität folgt, dass jede Region von einem Dreieck berandet ist [KN12, Kap. 12.3]. Der *Dualgraph* eines maximal planaren Graphen hat somit nur Knoten vom Grad drei. Diese Eigenschaft machen wir uns beim Zeichnen in Abschnitt 2.2 zu nutzen.

Eine *Rechtecksdarstellung* besteht aus der Zuordnung jedes Knotens $v \in V$ zu einem Rechteck $r_v \in R$ durch die Funktion $r : V \rightarrow R$ und zur Fläche $a_v \in A$ durch $a : V \rightarrow A$, wobei jedes Rechteck r_v die Fläche a_v besitzen soll. Die Rechtecke $r(u)$ und $r(v)$ mit $u, v \in V$ sind genau dann adjazent zueinander, wenn die Kante uv in E liegt.

Im Folgenden beschränken wir uns auf diejenigen Rechtecksdarstellungen, die einen *maximal außenplanaren Graphen* besitzen.

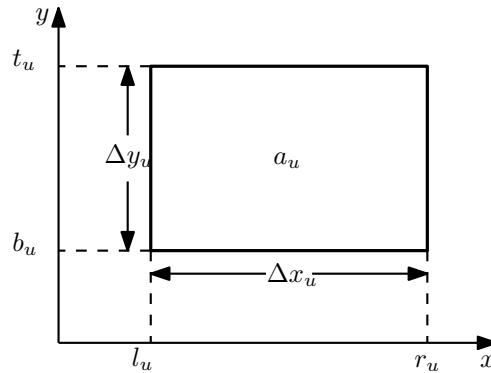


Abb. 2.2.: Rechteck u

Ein Rechteck $r(u)$ mit $u \in V$ des dualen Graphen $G = (V, E)$ hat einen unteren Rand mit der Ordinate b_u , einen oberen Rand t_u , einen linken Rand mit der Abszisse l_u sowie einen rechten Rand r_u . Außerdem besitzt es die Fläche $a_u = \Delta x_u \cdot \Delta y_u$ und das Seitenverhältnis $ratio_u = \Delta y_u / \Delta x_u$ mit $\Delta x_u = r_u - l_u$ und $\Delta y_u = t_u - b_u$, siehe dazu Abb. 2.2. Das Rechteck u kann man somit auch als kartesisches Produkt $[l_u, r_u] \times [b_u, t_u]$ schreiben.

2.2. Zeichnen der Rechtecksdarstellung

Der folgende Algorithmus zeichnet die Rechtecksdarstellung des maximal außenplanaren Graphen $G = (V, E)$. Zuerst wird in `MAXOUTERPLANARRECTANGULAR(G)` (Alg. 1) durch `INITIALIZE()` der duale Graph $G^* = (V^*, E^*)$ zu G erstellt. Im Punkt $v^* \in V^*$ treffen sich jeweils drei Ecken der Rechtecke aus R . Die Ecken der Punkte u^* und v^* sind über eine Rechtecksseite miteinander verbunden, wenn $\{u^*, v^*\} \in E^*$. Die Warteschlange Q speichert die besuchten und noch nicht abgeschlossenen Facetten von G . In `START()` werden die ersten beiden Rechtecke gesetzt, die die Basis des weiteren Verfahrens bilden. Solange noch Facetten in Q enthalten sind, werden diese bearbeitet. Dabei wird das in dieser Facette noch nicht gesetzte Rechteck des Knotens w gewählt und an die Rechtecke der Knoten u und v mittels `PLACE()` platziert. Abschließend werden alle noch nicht

Algorithmus 1: MAXOUTERPLANARRECTANGULAR(G)

Eingabe : maximal äußerplanarer Graph $G = (V, E)$
Ausgabe : korrekte Rechtecksdarstellung von G
/* Initialisiere Graphen und gib den Dualgraph von G zurück */
1 $G^* = \text{INITIALIZE}(G)$
2 $Q = \text{new Queue}()$
/* Setze die ersten beiden Rechtecke */
3 $\text{START}(G, G^*, Q)$
4 **while not** $Q.\text{empty}()$ **do**
5 $u^* = Q.\text{dequeue}()$
6 $w \in V$ ist der Knoten des noch nicht gesetzten Rechtecks der Facette von u^*
7 $u, v \in V$ sind die Knoten der bereits gesetzten Rechtecke der Facette von u^*
8 $\text{PLACE}(u, v, w)$
9 $\text{visited}[u^*] = \text{true}$
 /* Füge unbesuchte Nachbarflächen von u^* in die Schlange */
10 **foreach** Nachbar v^* von u^* **do**
11 | **if** $\text{visited}[v^*] == \text{false}$ **then** $Q.\text{enqueue}(v^*)$

Algorithmus 2: INITIALIZE(G)

Eingabe : maximal äußerplanarer Graph G
Ausgabe : Dualgraph von G
1 erstelle Dualgraph $G^* = (V^*, E^*)$ aus G
2 **foreach** Knoten $v^* \in V^*$ **do** $\text{visited}[v^*] = \text{false}$
3 **foreach** Knoten $v \in V$ **do** $\text{placed}[v] = \text{false}$
4 **return** G^*

besuchten Nachbarfacetten zur Warteschlange hinzugefügt.

Bei $\text{INITIALIZE}(G)$ (Alg. 2) wird aus dem Graphen $G = (V, E)$ der duale Graph $G^* = (V^*, E^*)$ gebildet. Außerdem wird jeder Knoten $v^* \in V^*$ als nicht besucht und $v \in V$ als nicht gesetzt markiert. Der duale Graph G^* wird am Ende übergeben.

In $\text{START}()$ (Alg. 3) wird zu Beginn die Außenfacette von G ermittelt. Von dieser aus wählt man eine äußere Kante e mittels $\text{DUAL}(e^*)$, welcher genau die zu e^* duale Kante liefert. Die Rechtecke der zu e inzidenten Punkte werden nebeneinander platziert. Zudem werden sie als gesetzt markiert und die zu u^* an e angrenzende, noch nicht bearbeitete, Dreieckfacette zur Warteschlange hinzugefügt.

Beim Platzieren des Rechtecks $r(w)$ in PLACE (Alg. 4) muss zuerst entschieden werden, ob es auf $r(u)$ oder $r(v)$ gesetzt wird. Danach wird die Breite Δx_w auf die Hälfte der noch nicht besetzten Länge der Seite des Rechtecks $r(u)$ bzw. $r(v)$ gesetzt. Hierbei gibt $\text{FREE}(u') = \Delta x_{u'} - \sum_{\{u', v'\} \in E, b_{v'} = t_{u'}} \Delta x_{v'}$ den noch freien Betrag der Seite $t_{u'}$ an. Nun muss das Rechteck von w nur noch entsprechend platziert werden, je nachdem ob es links oder rechts der Rechtecke von u oder v liegt.

Algorithmus 3: START(G, G^*, Q)

Eingabe : maximal äußerplanarer Graph G , Dualgraph G^* , Warteschlange Q

```
1 wähle  $u^* \in V^*$  mit  $\text{deg}(u^*)$  ist maximal // äußere Fläche in  $G$ 
2 visited[ $u^*$ ] = true
3  $e = \text{DUAL}(\{u^*, v^*\})$  mit  $u^*$  adjazent zu  $v^* \in V^*$  und  $e \in E$ 
4 setze die Rechtecke der Punkte  $u$  und  $v$  mit  $e = \{u, v\}$  so nebeneinander,
   dass  $b_u == b_v$  und  $\Delta x_u == \Delta x_v$  gilt
5 if  $\Delta y_u == \Delta y_v$  then
6   |  $\Delta x_u = \Delta x_u + \epsilon$  // falls die Rechtecke gleiche Höhe haben
7 placed[ $u$ ] = true
8 placed[ $v$ ] = true
9 Q.enqueue( $v^*$ )
```

Algorithmus 4: PLACE(u, v, w)

Eingabe : Gesetzte Rechtecke der Punkte u und v , nicht gesetztes Rechteck $r(w)$

```
1 if  $t_u < t_v$  // ist  $u$  niedriger als  $v$ 
2 then
3   | setze  $b_w$  an  $t_u$  // setze  $w$  auf  $u$ 
4   | setze  $\Delta x_w = \frac{1}{2} \text{FREE}(u)$  // Breite von  $w$  = halbe freie Seite von  $u$ 
5   | if  $l_u < l_v$  // falls  $u$  links neben  $v$ 
6   | then
7   | | setze  $r_w$  an  $l_v$  // setze  $w$  von links an  $v$ 
8   | else
9   | | setze  $l_w$  an  $r_v$  // setze  $w$  von rechts an  $v$ 
10  | if  $t_w == t_v$  // falls  $w$  und  $v$  gleich hoch
11  | then  $\Delta x_w = \Delta x_w - \epsilon$ 
12 else
13  | setze  $b_w$  an  $t_v$  // setze  $w$  auf  $v$ 
14  | setze  $\Delta x_w = \frac{1}{2} \text{FREE}(v)$  // Breite von  $w$  = halbe freie Seite von  $v$ 
15  | if  $l_u < l_v$  // falls  $u$  links neben  $v$ 
16  | then
17  | | setze  $l_w$  an  $r_u$  // setze  $w$  von rechts an  $u$ 
18  | else
19  | | setze  $r_w$  an  $l_u$  // setze  $w$  von links an  $u$ 
20  | if  $t_w == t_u$  // falls  $w$  und  $u$  gleich hoch
21  | then  $\Delta x_w = \Delta x_w - \epsilon$ 
```

2.3. Beweis der Korrektheit für maximal außenplanare Graphen

Gegeben sei ein Graph $G = (V, E)$. Zu diesem soll ein korrektes Rechteckslayout erstellt werden, wie es unter Abschnitt 2.1 definiert wurde. Die Korrektheit wird induktiv über die Anzahl der gesetzten Rechtecke bewiesen.

Ausgehend von einer Rechtecksdarstellung des Graphen $G_k = (V_k, E_k)$ mit k Rechtecken und $V_k \subseteq V$ sowie $E_k = \{(u, v) \in E \mid u, v \in V_k\}$, gilt bei jedem Schleifendurchlauf folgende Invariante:

Es gibt eine freie Ecke für das als nächstes zu setzende Rechteck $r(w)$ mit $w \in V \setminus V_k$, sodass es adjazent zu seinen Vorgängern $r(u)$ und $r(v)$ ist, mit $u, v \in V_k$.

Die Punkte u und v sind genau diejenigen, welche in V_k enthalten sind und deren Rechtecke beim Durchlaufen der einzelnen Facetten des Graphen von G zuerst gesetzt wurden.

(I) Induktionsanfang:

Wir starten mit einer äußeren Kante $\{u, v\} \in E$. Die beiden Rechtecke $r(u)$ und $r(v)$ werden gemäß dem Algorithmus nebeneinander angeordnet und die Knoten $u, v \in V$ zur Menge V_2 hinzugefügt. Da u und v keine Vorgänger besitzen, müssen sie auch nicht adjazent zu diesen sein und können frei auf der Fläche angeordnet werden. Hierdurch ergibt sich eine freie Ecke, an der das zu $r(u)$ und $r(v)$ adjazente Rechteck $r(w)$ mit $w \in V$ Platz findet; siehe Abb. 2.3. Da jede Facette in G von einem Dreieck umgeben ist, existiert nur genau ein solches Rechteck $r(w)$. Die Invariante ist damit für $k = 2$ erfüllt.

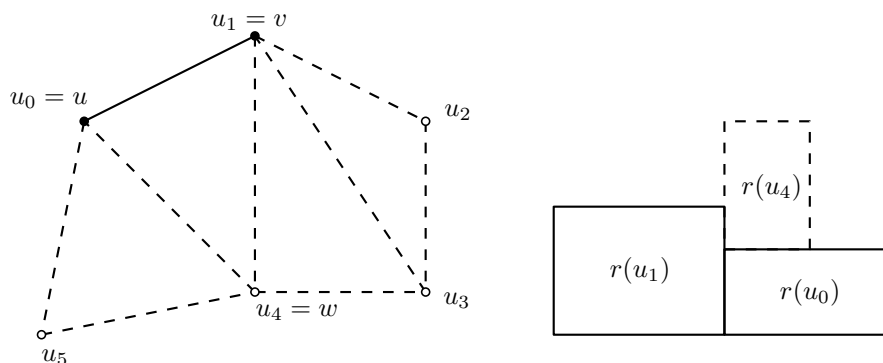


Abb. 2.3.: Setzen der ersten beiden Rechtecke

(II) Induktionsvoraussetzung:

Es existiert eine Ecke im Graphen G_k , an die das als nächstes zu setzende Rechteck $r(w)$ mit $w \in V \setminus V_k$ gesetzt werden kann. Der Knoten w muss nach Invariante adjazent zu zwei schon gesetzten Knoten $u, v \in V_k$ sein; siehe Abb. 2.4.

(III) Induktionsschritt:

Im Graphen G_k tritt man über die Kante $\{u, v\}$ in die dreieckige Facette $\{u, v, w\}$. Aufgrund der Induktionsvoraussetzung gibt es eine freie Ecke, an der man das Rechteck w anbringen kann. Hierbei ist w zu u und v adjazent. Eines der beiden Rechtecke u und v hat eine obere Kante t welche oberhalb der des anderen

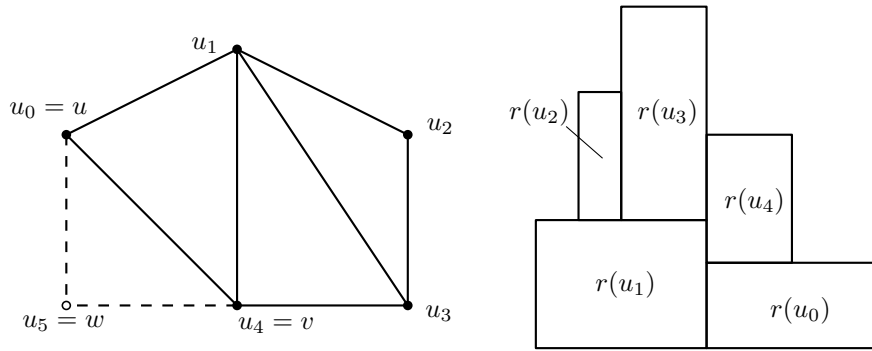


Abb. 2.4.: G_5 mit fünf gesetzten Rechtecken

Rechtecks liegt. O.B.d.A. wird davon ausgegangen, dass t_u oberhalb von t_v liegt, womit $t_u > t_v$ gilt. Außerdem liegt $r(u)$ entweder an bzw. auf der linken oder rechten Seite von $r(v)$. Wir gehen davon aus, dass es links liegt und damit $r_u < r_v$ gelten muss; siehe Abb. 2.5.

Die untere Kante b von w wird auf t_v gesetzt, so dass $\Delta y_w = \frac{1}{2} \text{FREE}(v)$ gilt. Außerdem soll, wie in Abb. 2.6 gezeigt, l_w auf r_u liegen. Der Knoten w wird letztendlich zur Menge V_{k+1} hinzugefügt.

Falls das gesetzte Rechteck w die selbe Höhe t_w hat, wie das zu ihm benachbarte Rechteck u , wird Δx_w verringert und somit Δy_w erhöht. So ist gewährleistet, dass immer zwei neue Ecken entstehen.

Durch dieses Verfahren ergeben sich zwei neue freie Ecken an den Rechtecken von w und u sowie w und v . Im Graphen G_{k+1} gibt es höchstens zwei Möglichkeiten einen nächsten Knoten aus G hinzuzufügen. Dieser ist zu w und entweder u oder v adjazent und kann an die entsprechenden Ecken angefügt werden. Die Invariante ist somit für $k+1$ erfüllt. Es existiert eine Rechtecksdarstellung mit $k+1$ Rechtecken; siehe Abb. 2.7.

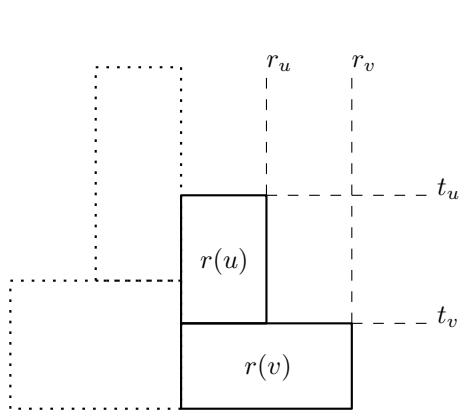


Abb. 2.5.: u_p und u_q vor dem Setzen von u_z

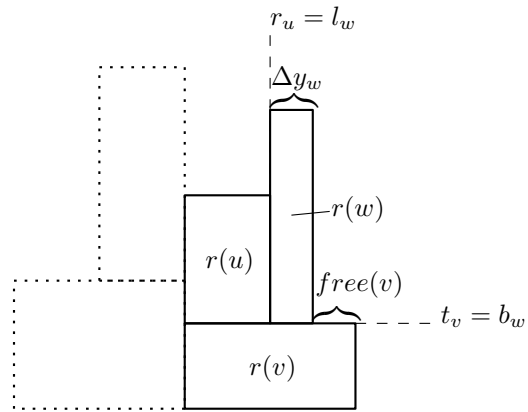


Abb. 2.6.: u_p und u_q nach dem Setzen von u_z

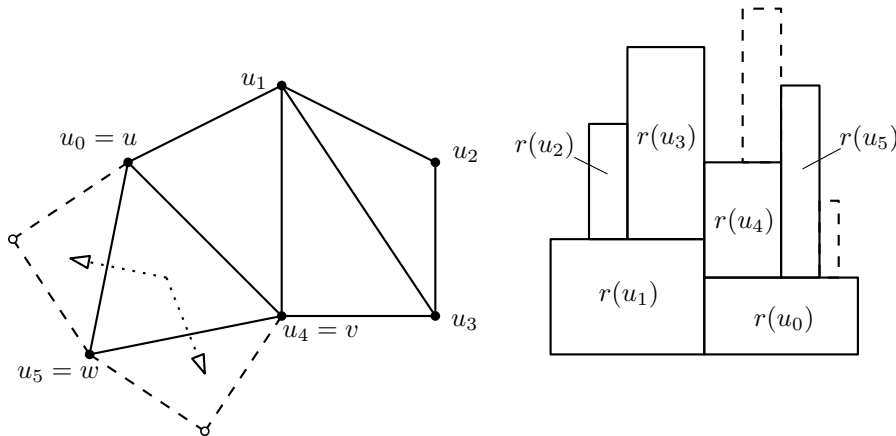


Abb. 2.7.: Weitere Möglichkeiten Knoten zu G' hinzuzufügen, nachdem u_z gesetzt wurde

2.4. Implementierung in Java

Um nicht nur die Korrektheit, sondern auch die Umsetzbarkeit des Algorithmus zu zeigen, wird dieser in Java implementiert und anhand eines Graphen getestet. Als Grundlage dient die Bibliothek *JGraphT* von Naveh [Nav]. Zu Beginn wird der zu zeichnende Graph *RoomGraph*, sowie sein Dualgraph *DualRoomGraph* in der Klasse *Graph* manuell erstellt; siehe Abbildung 2.8.

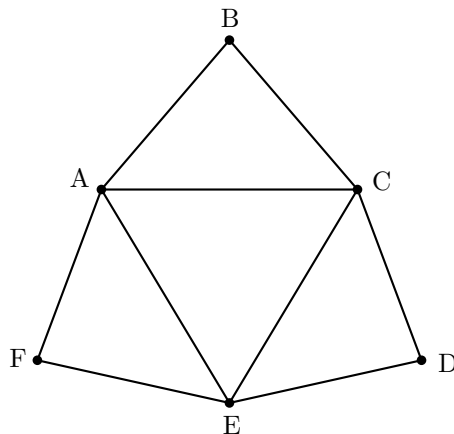


Abb. 2.8.: Der als Rechtecksdarstellung zu zeichnende Graph

Als Knoten des *RoomGraph* dienen Objekte der Klasse *RoomNode*, welche die Größe, den Namen und das zugehörige Rechteck speichern. Der Dualgraph besitzt Knoten der Klasse *FaceNode*, wobei lediglich die zur Facette angrenzenden *RoomNodes* festgehalten werden. Diese beiden Graphen werden an die statische Klasse *MaxOuterPlanarToRectangle* mithilfe der Funktion *toRectangle(Graph, Dualgraph)* übergeben. Diese verfährt wie in Abschnitt 2.2 und ruft zuerst die private Funktion *initialize()* auf. Dabei wird zur Vereinfachung auf die automatische Generierung des Dualgraphen verzichtet und der zuvor

übergebene verwendet. Zur Markierung, ob eine Facette besucht oder der Knoten eines Rechtecks gesetzt wurde, verwenden wir *HashMaps*. Eine *LinkedList* dient als Schlange zur Speicherung der zu besuchenden Facetten und eine weitere zum Festhalten der erstellten Rechtecke. Die Funktion *start()* ist analog zum Algorithmus 3 verfasst, wobei die erstellten Rechtecke als Objekte der Klasse *Rectangle* realisiert werden. Dabei hält dieses den unteren linken und den oberen rechten Eckpunkt, was wiederum ein Objekt der Klasse *Point* mit zwei Koordinaten darstellt. Nachfolgend werden wie in Algorithmus 4 die restlichen Rechtecke gesetzt. Dieser ist in einen Unteralgorithmus *placeOnTop(u,v,w)* aufgeteilt, um eine Codeduplizierung bei der Unterscheidung des größeren zum kleineren Rechteck zu vermeiden. Bei der Umsetzung gestaltete sich vor allem die Funktion *dual()*, welche im Algorithmus 3 benötigt wird, als unerwartet umfangreich, da JGraphT zur Handhabung von Dualgraphen keine entsprechende Lösung liefert.

Die Ausgabe erfolgt über die von Martin Fink und Philipp Kindermann bereitgestellte Klasse zur Erstellung von Ipe Dateien unter Java. Das von Ottfried Cheong entwickelte Vektorzeichenprogramm *Ipe extensible drawing editor* [Che] verwendet als Grundlage für Zeichnungen XML Dateien. Diese eignen sich insbesondere zur Verwendung in PDF Dokumenten. Die ins PDF Format umgewandelte Ausgabe des Java Programms ist in Abbildung 2.9 dargestellt. Zur Übersichtlichkeit wurde die Beschriftung der Rechtecke verkürzt.

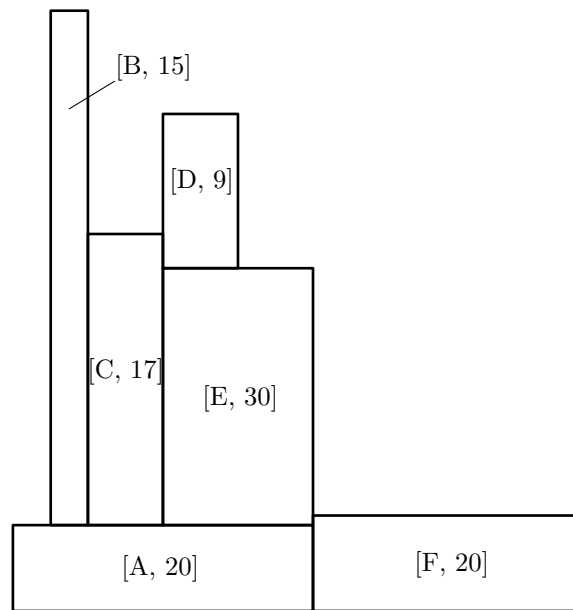


Abb. 2.9.: Die Rechtecksdarstellung als Ausgabe des Java Programms mit der Beschriftung [Name, Größe]

3. Außenplanare Graphen

Das vorherige Kapitel 2 beschäftigt sich mit maximal außenplanaren Graphen, da sie aufgrund ihrer dreieckigen Facetten effektiv zu durchlaufen sind. Diese Einschränkung soll im Folgenden auf die Klasse der außenplanaren Graphen erweitert werden. Wie schon in Abschnitt 2.1 beschrieben, ist ein Graph genau dann außenplanar, wenn eine Einbettung in die Ebene existiert, bei der alle Knoten am Rand eines gemeinsamen Gebiets liegen. Insbesondere heißt dies im Unterschied zu maximal außenplanaren Graphen, dass vier- und mehreckige Facetten existieren können, ohne die Außenfacette. Der Algorithmus aus Abschnitt 2.2 ist jedoch nur in der Lage Dreiecksfacetten zu durchlaufen, da hier die Platzierung des noch nicht gesetzten Rechtecks durch die beiden schon gezeichneten eindeutig festgelegt ist.

3.1. Erweiterung des Ausgangsgraphen

Um die Einschränkung der Maximalität zu umgehen, sollte überlegen werden welche Bedeutung sie für die Rechtecksdarstellung hat. Ausgehend von einem maximal außenplanaren Graphen erhält man einen nicht maximalen außenplanaren Graphen, indem eine Kante im Inneren entfernt wird. Dadurch werden zwei Facetten zu einer verschmolzen. Für die Darstellung bedeutet das eine Veränderung der Eigenschaften von genau zwei Rechtecken. Diese beiden sind nicht mehr adjazent zueinander und müssen durch eine freie Fläche voneinander getrennt werden. Da die nun nicht mehr dreieckige Facette des Graphen von Kanten umgeben ist, kann die zu ihr gehörende Fläche nicht außerhalb der Rechtecksdarstellung liegen und bildet somit ein Gebiet im Inneren; siehe Abbildung 3.1. Führt man dieses Vorgehen fort, so wird die Fläche vergrößert oder es entstehen neue freie Flächen.

Eine Besonderheit entsteht beim Entfernen von Kanten, die die Außenfacette begrenzen. Dabei wird das Gebiet, welches die Darstellung umgibt, vergrößert und dringt zwi-

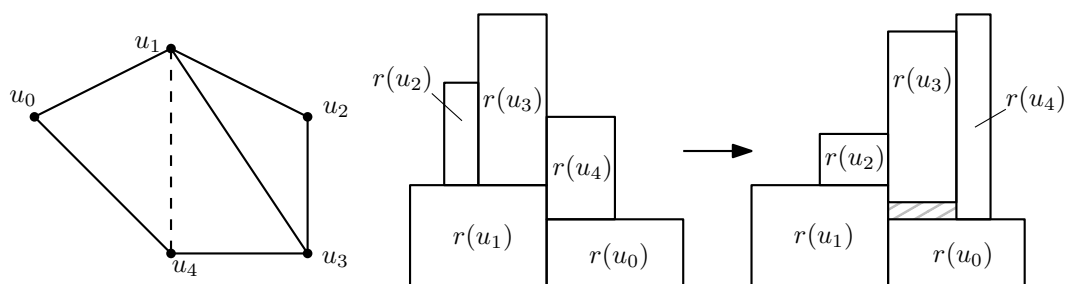


Abb. 3.1.: Auswirkung des Entferneins einer Kante auf die Rechtecksdarstellung

schen zwei zuvor benachbarte Rechtecke. Im Extremfall werden die Rechtecke komplett voneinander getrennt. O.B.d.A. beschäftigen wir uns im Folgenden ausschließlich mit zusammenhängenden Graphen. Besteht ein Graph aus mehreren Zusammenhangskomponenten, so können deren Rechtecksdarstellungen unabhängig voneinander in der Ebene platziert werden.

3.2. Schematischer Ablauf der Zeichnung

Um eine Zeichnung zu erhalten, die alle Gebiete des Graphen erreicht und die nicht dreiecksförmigen Facetten richtig darstellt, muss der Graph um Knoten und Kanten erweitert werden. Diese helfen beim Durchlaufen des Graphen und bei der Zeichnung freizulassender Bereiche. Anschließend wird der Graph mittels eines zu Kapitel 2.2 angepassten Algorithmus gezeichnet, wobei auch die freizulassenden Bereiche als Rechtecke in die Darstellung eingefügt werden. Diese werden abschließend aus der Zeichnung entfernt.

3.2.1. Einfügen von Dummy-Knoten

Zu Beginn liegt ein außenplanarer Graph $G = (V, E)$ mit Knoten $v \in V$ und Kanten $e \in E$ vor. Die Rechtecksdarstellung sei analog zu Abschnitt 2.1 definiert. Damit der Algorithmus aus Abschnitt 2.2 an die neue Graphenklasse angepasst werden kann, müssen wir den Ausgangsgraphen entsprechend erweitern. Dies geschieht mithilfe von *Dummy-Knoten*. Sie stehen stellvertretend für rechteckige Gebiete in der Darstellung, die mehrere Rechtecke voneinander trennen.

Dabei wird zwischen *inneren* und *äußeren* Dummy-Knoten unterschieden. Die Inneren $v \in V_i$ sind komplett von Kanten des Graphen G umgeben und werden entsprechend der Rechtecksdarstellung durch die Funktion $r : V_i \rightarrow R_f$ auf freie Rechteckflächen R_f abgebildet, die von anderen Rechtecken umgeben sind. Für jede nicht dreieckige Facette, ohne die Außenfacette, wird ein innerer Dummy-Knoten v zur Menge V_i hinzugefügt. Der Knoten v wird durch Kanten $uv \in E_i$ mit Knoten u verbunden, die die zu v gehörige Facette bilden; siehe Knoten X in Abbildung 3.2.

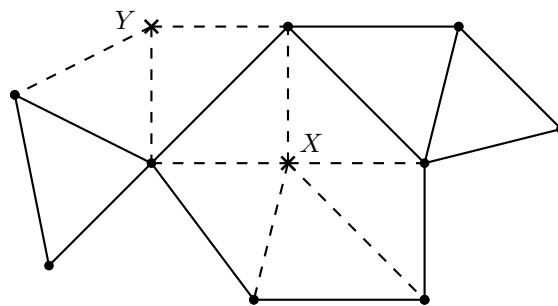


Abb. 3.2.: Der außenplanare Graph G (durchgezogene Linien und Punkte) und seine Erweiterung G' (gestrichelte Linien und Kreuze)

Um die Zugehörigkeit der äußeren Dummy-Knoten zu bestimmen, müssen wir zuerst die Begriffe *Zusammenhang* und *A-B-Trenner* definieren. Ein Graph $G = (V, E)$ heißt zusammenhängend, wenn jedes Knotenpaar $v, w \in V$ durch einen Weg aus Kanten $e \in E$ miteinander verbunden werden kann. Darauf aufbauend wird ein *A-B-Trenner* in Diestel [Die00, Kap. 0.4] wie folgt definiert: Seien $A, B \subseteq V$ Teilmengen der Knoten und $X \subseteq V \cup E$ eine Teilmenge der Knoten und Kanten des Graphen $G = (V, E)$. Enthält jeder *A-B-Weg* in G eine Ecke oder Kante aus X , so trennt X die Mengen A, B in G und ist ein *A-B-Trenner*; insbesondere gilt dann $A \cap B \subseteq X$.

Da wir, wie in Abschnitt 3.1 beschrieben, nur zusammenhängende Graphen betrachten, wird jedem *A-B-Trenner* mit $X \subseteq V$ und $|X| = 1$ ein äußerer Dummy-Knoten v zugeordnet, in Zeichen X_v , und zur Menge V_a hinzugefügt. Zudem wird die Kante vx , mit $x \in X_v$, zur Kantenmenge E_a hinzugefügt. Auch die beiden Kanten va und vb sollen in E_a sein, mit $a \in A$ und $b \in B$, sowie a, b sind adjazent zu X_v . Außerdem müssen die beiden Kanten ax und bx mit $x \in X_v$ zur Außenfacette benachbart sein; siehe Knoten Y in Abbildung 3.2. Die äußeren Dummy-Knoten werden mittels $r : V_a \rightarrow R_f$ auf diejenigen Rechtecksflächen projiziert, die direkt an die Außenfacette der Rechtecksdarstellung angrenzen. Folgen mehrere *A-B-Trenner* aufeinander, so werden die äußeren Dummy-Knoten abwechselnd auf die jeweils gegenüberliegende Seite gesetzt, um die Planarität des Graphen zu erhalten.

Fügt man die Dummy-Knoten zum Graphen G hinzu, so erhalten wir den Graphen $G' = (V', E')$. Die Knotenmenge V' besteht aus den Ursprungsknotenmenge V , sowie den äußeren und inneren Dummy-Knotenmengen V_a und V_i , also $V' = V \cup V_a \cup V_i$. Die Menge der Kanten $E' = E \cup E_a \cup E_i$ ist ebenso aus den ursprünglichen sowie den neu hinzugefügten Kanten zusammengesetzt.

3.2.2. Durchlaufen des Graphen

Durch die Vorarbeit in Unterabschnitt 3.2.1 wird ein erfolgreiches Durchlaufen des Graphen G' gewährleistet. Die inneren Dummy-Knoten V_i teilen die λ -eckigen Facetten in kleine Dreiecke auf. Die äußeren Dummy-Knoten V_a fügen einen Weg zu zwei Teilgraphen hinzu, die nur über einen Knoten miteinander verbunden sind. Zudem dienen diese beiden als Hinweis auf einen Spezialfall im Ablauf des in Abschnitt 2.2 beschriebenen Algorithmus.

Als Eingabe dient der erweiterte Ausgangsgraph G' , wobei die in Algorithmus 3 gewählte äußere Kante $e = \{u, v\}$ in $E = E' \setminus \{E_a \cup E_i\}$ liegen muss. Im Algorithmus 1 werden nach der Wahl der als nächstes zu bearbeitenden Knoten (Zeile 6-7) drei Fälle unterschieden:

Handelt es sich bei dem noch nicht gesetzten Knoten $w \in V'$ um einen der Menge V zugehörigen, so folgt man dem ursprünglichen Vorgehen des Algorithmus. Ist $w \in V_i \cup V_a$, wird entsprechend äußerer oder innerer Knoten eines der folgenden Verfahren angewendet. Hat man den gesamten Graphen durchlaufen, müssen die Rechtecke der Dummy-Knoten nur noch aus der Rechtecksdarstellung entfernt werden, um eine korrekte Darstellung eines außenplanaren Graphen zu erhalten.

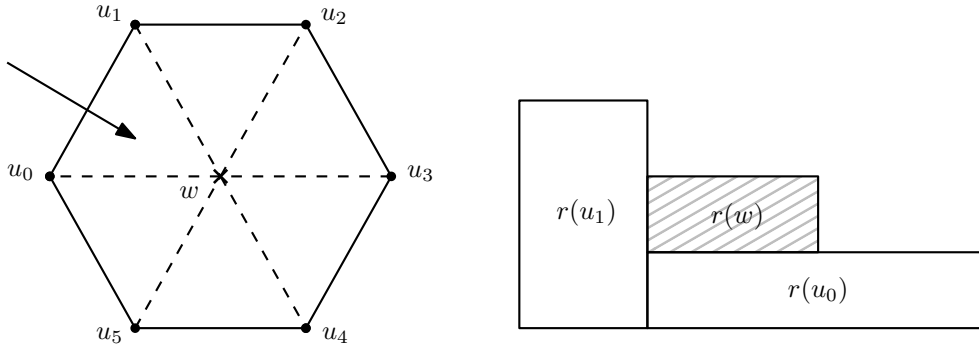


Abb. 3.3.: Bestimmung des freien Bereichs als erster Schritt beim Durchlaufen einer λ -eck-Facette

Ein innerer Dummy-Knoten

Hierbei wird die gesamte λ -eckige Facette in einem Schritt umlaufen, die $w \in V_i$ umgibt. Diese besteht aus den Knoten $\{u_0, u_1, \dots, u_{\lambda-1}\}$, die im Uhrzeigersinn fortlaufend nummeriert sind. Zunächst tritt man über eine Kante $\{u, v\} = \{u_0, u_1\}$ in die Dreiecksfacette mit den Eckpunkten $\{u_0, u_1, w\}$. Nun muss als erstes das zum Dummy-Knoten gehörige Rechteck $r(w)$ auf $r(u_0)$ bzw. $r(u_1)$ gesetzt werden. O.B.d.A. gehen wir davon aus, dass $r(u_0)$ das kleinere von beiden Rechtecken ist und somit $t_{u_0} < t_{u_1}$ gilt. Zudem liege u_1 links auf bzw. neben u_0 . Unter dieser Annahme können wir $r(w)$ auf $r(u_0)$ setzen, mit $b_w = t_{u_0}$ und $\Delta x_w = 1/2 \cdot \text{FREE}(u_0)$. Außerdem soll die obere Kante von $r(w)$ unterhalb der oberen Kante von $r(u_1)$ liegen, sodass $t_w = 1/2 \cdot (t_{u_1} + t_{u_0})$ gilt. Somit wäre die frei zu lassende Fläche in die Rechtecksdarstellung eingefügt, wie in Abbildung 3.3 zu sehen, und wir können uns den sie umgebenden Rechtecken widmen.

Die zum Rechteck $r(w)$ adjazenten Dreiecksfacetten durchläuft man der Nummerierung entsprechend im Uhrzeigersinn. Hierbei wird das in der Darstellung noch nicht platzierte Rechteck $r(u_\alpha)$ mit $\alpha \in \{2, \dots, \lambda-3\}$ auf $r(w)$ mit $b_{u_\alpha} = t_w$ und $\Delta x_{u_\alpha} = 1/2 \cdot \text{FREE}(w)$ gesetzt. Zudem soll es rechts neben $r(u_{\alpha-1})$ liegen, mit $l(u_\alpha) = r(u_{\alpha-1})$. Die

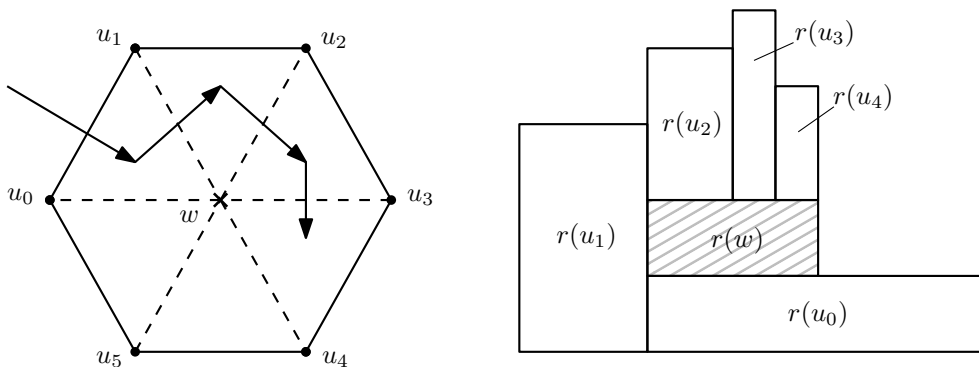


Abb. 3.4.: Die Rechtecke werden auf der freien Fläche bündig platziert

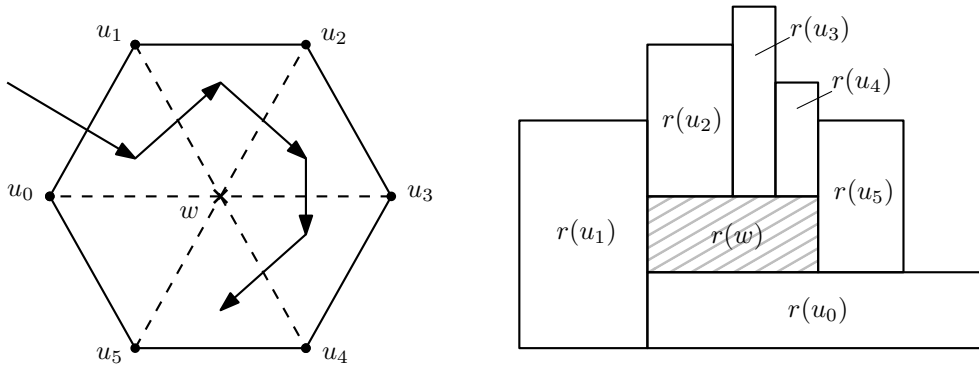


Abb. 3.5.: Fertige Rechtecksdarstellung einer λ -eck-Facette

Höhe t_{u_α} ergibt sich aus der Fläche $a(u_\alpha)$. Ist man beim Rechteck $r(u_{\lambda-2})$ angekommen, so setzt man dieses wie beschrieben auf $r(w)$, allerdings gilt diesmal $\Delta x_{u_{\lambda-2}} = \text{FREE}(w)$; siehe Abbildung 3.4.

Beim letzten Knoten $u_{\lambda-1}$ setzen wir das Rechteck rechts neben $r(w)$. Es gilt $b_{u_{\lambda-1}} = t_{u_0}$ und $l_{u_{\lambda-1}} = r_w$. Die Breite Δx von $u_{\lambda-1}$ wird auf $1/2 \cdot \text{FREE}(u_0)$ gesetzt. Die Höhe $t_{u_{\lambda-1}}$ ergibt sich wiederum aus der Fläche $a(u_{\lambda-1})$. Gilt dabei $t_{u_{\lambda-1}} \leq b_{u_{\lambda-2}}$, so verringern wir Δx von $u_{\lambda-1}$ soweit, bis $t_{u_{\lambda-1}} > b_{u_{\lambda-2}}$ gilt; siehe Abbildung 3.5.

Damit sind alle Rechtecke der zur λ -eck-Facette gehörenden Knoten miteinander benachbart, jedoch durch das innen liegende Rechteck $r(w)$ voneinander getrennt. Alle zum λ -Eck benachbarten unbesuchten Dreiecksfacetten müssen zur Schlange Q hinzugefügt werden, um sie später weiter bearbeiten zu können. Außerdem müssen die einzelnen Facetten nach der Bearbeitung als besucht markiert werden, damit das λ -Eck zu einem späteren Zeitpunkt nicht noch einmal durchlaufen wird.

Ein äußerer Dummy-Knoten

Liegt ein äußerer Dummy-Knoten $w \in V_a$ vor, so werden die beiden angrenzenden Dreiecksfacetten durchlaufen, wobei der Knoten v ein A - B -Trenner sei. Dieser wird als Recht-

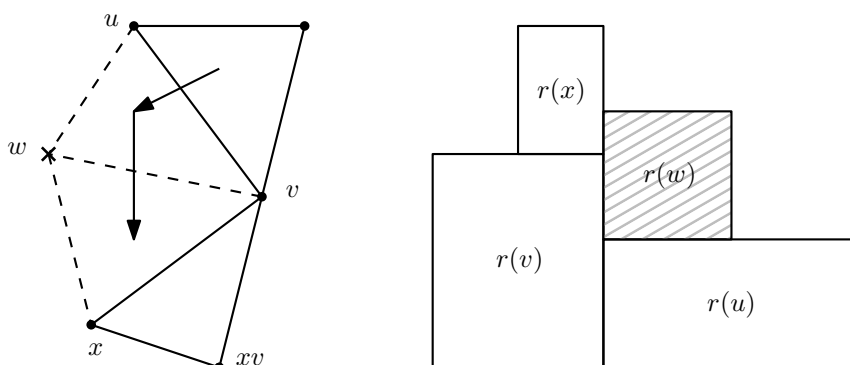


Abb. 3.6.: Der äußere Dummy-Knoten kann wie ein normaler Knoten behandelt werden

eck in die Darstellung eingefügt. Danach kann wie im Algorithmus beschrieben fortgefahren werden.

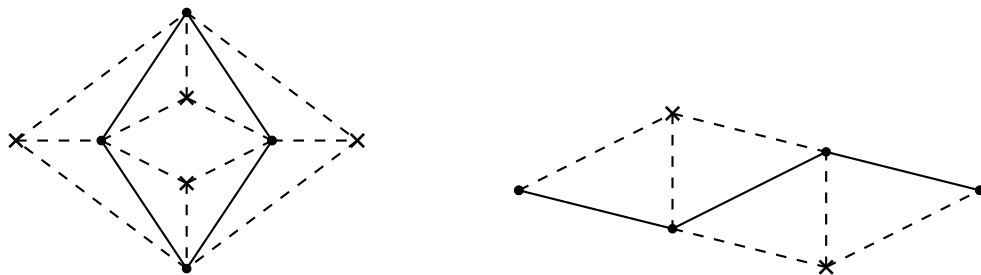
Als erstes wird über die Kante $\{u, v\}$ die Dreiecksfacette $\{u, v, w\}$ erreicht. Dabei sind $r(u)$ und $r(v)$ schon gesetzt, weshalb nur noch $r(w)$ gezeichnet werden muss. Dieses kann wie in Algorithmus 4 gesetzt werden. Die Größe setzt man auf $a(w) = 1$, da sie keine Rolle spielt. Somit ist die Facette $\{u, v, w\}$ abgeschlossen und das zu $r(v)$ und $r(w)$ adjazente Rechteck $r(x)$ kann an die neu entstandenen Ecken gesetzt werden. Dies geschieht wieder nach Algorithmus 4. Damit ist $r(w)$ zu $r(u)$ und $r(v)$, sowie $r(x)$ zu $r(w)$ und $r(v)$ adjazent. Die Rechtecke $r(u)$ und $r(x)$ sind jedoch durch $r(w)$ voneinander getrennt; siehe Abbildung 3.6. Die benachbarte Facette wird zur Schlange Q hinzugefügt. Zudem muss man darauf achten, dass die an den Dummy-Knoten angrenzenden Dreiecksfacetten als besucht markiert werden.

3.3. Beweis der Korrektheit für außenplanare Graphen

Die Korrektheit des in Abschnitt 3.2 beschriebenen Verfahrens geschieht induktiv über die Anzahl der gesetzten Rechtecke in der Darstellung. Dabei stützen wir uns auf den in Abschnitt 2.3 erbrachten Beweis. Anfänglich sei ein Graph $G = (V, E)$ gegeben, den wir entsprechend Abschnitt 3.2.1 in $G' = (V', E')$ umwandeln. Dies ist konstruktionsbedingt möglich und korrekt ausführbar. Der Graph $G_k = (V_k, E_k)$ entspricht der Darstellung mit k gesetzten Rechtecken und ist ein Teilgraph von G' . Hierbei gilt $V_k \subseteq V'$ und $E_k = \{(u, v) \in E' \mid u, v \in V_k\}$. Die Invariante lautet:

Es gibt eine freie Ecke für das als nächstes zu setzende Rechteck $r(w)$ mit $w \in V' \setminus V_k$, sodass es adjazent zu seinen Vorgängern $r(u)$ und $r(v)$ ist, mit $u, v \in V_k$.

Dabei seien $r(u)$ und $r(v)$ die schon gesetzten und $r(w)$ das als nächstes zu setzende Rechteck. Der Beweis kann für dreieckige Facetten ohne Dummy-Knoten übernommen werden, da diese wie bei maximal außenplanaren Graphen behandelt werden. Der Spezialfall, dass die Facette einen Dummy-Knoten enthält, wird pro Schleifendurchlauf bewiesen. Dabei fügt man mehrere Rechtecke auf einmal zu G_k hinzu.



- (a) Bilden die A - B -Trenner einen Kreis, so sind sie keine A - B -Trenner mehr. Die vorliegende Darstellung ist nie möglich.
- (b) Bilden die A - B -Trenner keinen Kreis, so existieren Knoten, die keine A - B -Trenner sind

Abb. 3.7.: Existenz von zur Außenfacette benachbarter Kanten $e \in E$

(I) **Induktionsanfang:**

Da in Algorithmus 3 nur zur Außenfacette benachbarte Kanten aus $E = E' \setminus \{E_a \cup E_i\}$ gewählt werden, folgt die Korrektheit analog aus dem Induktionsanfang I in Abschnitt 2.3.

Angenommen eine solche Kante existiere nicht, so wären alle Kanten $e \in E$ von Kanten $e_a \in E_a$ umgeben. Das heißt der Graph bestünde vollständig aus A - B -Trennern, denen je ein Dummy-Knoten zugeordnet würde. Um die Kanten aus E_a den A - B -Trennern, deren Nachbarn, sowie dem Dummy-Knoten richtig zuzuordnen, müssten die A - B -Trenner einen Kreis bilden, wodurch sie keine A - B -Trenner mehr wären; siehe Abbildung 3.7a. Bilden diese keinen Kreis, so sind die äußersten Knoten keine A - B -Trenner, wie in Abbildung 3.7b zu sehen, was im Widerspruch zur Annahme steht. Also existiert immer eine zur Außenfacette benachbarte Kante $e \in E$. Die Invariante ist für $k = 2$ erfüllt.

(II) **Induktionsvoraussetzung:**

Es existiert eine Ecke im Graphen G_k , an die das als nächstes zu setzende Rechteck $r(w)$ mit $w \in V' \setminus V_k$ gesetzt werden kann. Der Knoten w muss nach Invariante adjazent zu zwei schon gesetzten Knoten $u, v \in V_k$ sein. Da $r(w)$ mit $w \in V_i \cup V_a$ rechteckig ist, gilt die Invariante auch für Dummy-Knoten.

(III) **Induktionsschritt:**

1. **Fall:** Eine Dreiecksfacette ohne Dummy-Knoten wird besucht.

Hier kann analog zu Induktionsschritt III in Abschnitt 2.3 vorgegangen werden.

2. **Fall:** Eine Dreiecksfacette mit innerem Dummy-Knoten wird besucht.

Da jeder geschlossenen, nicht dreieckigen Facette, außer der Außenfacette, genau ein Dummy-Knoten zugeordnet wird, müssen wir nur diesen Fall behandeln. Im Graphen tritt man dabei über die Kante $\{u_0, u_1\}$ in die λ -eckige Facette mit den Knoten $\{u_0, u_1, \dots, u_{\lambda-1}\}$, sowie Dummy-Knoten w ein. Nach Induktionsvoraussetzung existiert eine Ecke, an die $r(w)$ angefügt werden kann.

Als nächstes setzt man die Rechtecke $\{u_2, \dots, u_{\lambda-2}\}$ nebeneinander auf $r(w)$. Da diese über eine Kante $e \in E' \setminus E_i \cup E_a$ miteinander verbunden sind, existieren möglicherweise unbesuchte Facetten, die an diese Kanten angrenzen. Die Höhe der Rechtecke wird jedoch so gewählt, dass je zwei Rechtecke $\{u_{\alpha-1}, u_\alpha\}$ mit $\alpha \in \{2, \dots, \lambda-2\}$ eine neue Ecke bilden. Somit kann ein zu diesen adjazentes Rechteck in die Rechtecksdarstellung eingefügt werden.

Beim Setzen des letzten Rechtecks $u_{\lambda-1}$ an $u_{\lambda-2}$ und u_0 entstehen zwei neue Ecken. An diese kann man die zu $u_{\lambda-2}$ und $u_{\lambda-1}$ bzw. u_0 und $u_{\lambda-1}$ adjazenten Rechtecke anfügen.

Beim Schleifendurchlauf entstehen $\lambda - 1$ neue Ecken, an die die noch nicht gesetzten Rechtecke der Nachbarfacetten des λ -Ecks angebracht werden können; siehe Abbildung 3.8. Der Graph G_k wird um $\lambda - 1$ Rechtecke zu $G_{k+\lambda-1}$ erweitert. Die Invariante ist somit für $k + \lambda - 1$ erfüllt.

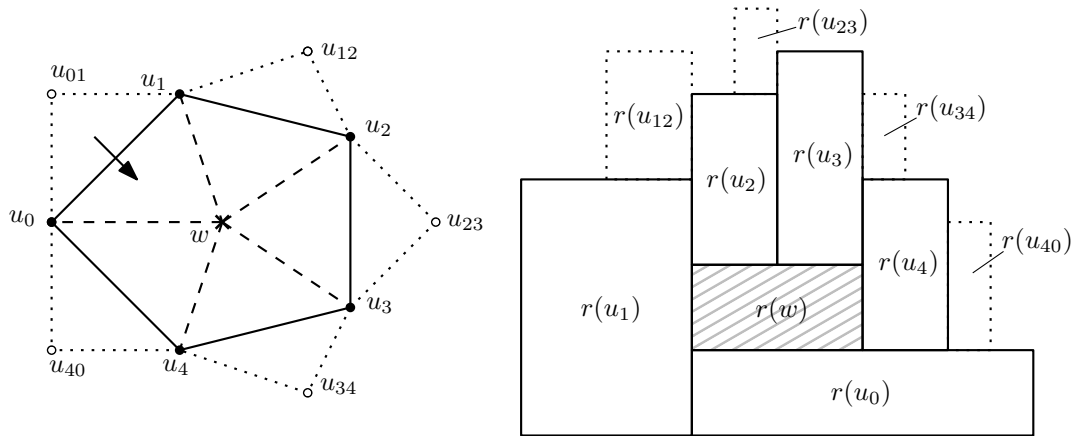


Abb. 3.8.: Das λ -Eck umgebende Rechtecke und ihre Platzierung in der Rechtecksdarstellung

3. Fall: Eine Dreiecksfacette mit äußerem Dummy-Knoten wird besucht.

Über die Kante $\{u, v\}$ mit dem A - B -Trenner v tritt man in die Dreiecksfacette $\{u, v, w\}$. Es existiert nach Induktionsvoraussetzung eine Ecke, an die das Rechteck $r(w)$ des Dummy-Knotens w angebracht werden kann. Da man über die Kante $\{u, v\}$ in das Dreieck $\{u, v, w\}$ tritt und eine weitere Kante $\{u, w\}$ zur Außenfacette benachbart ist, existiert nur eine Kante, die an ein noch nicht besuchtes Dreieck grenzt. Dies ist konstruktionsbedingt die Kante $\{v, w\}$.

Der zu w und v adjazente Knoten sei x . Da beim Platzieren von $r(w)$ auf Algorithmus 4 und somit das in Fall 1 benutzte Verfahren zurückgegriffen wird, existieren zwei neue Ecken, an die $r(x)$ gesetzt werden kann. Eine besteht dabei aus den Rechtecken $r(u)$ und $r(w)$, die andere aus $r(v)$ und $r(w)$. Das Rechteck $r(x)$ wird somit an letztere angebracht. Also existiert eine Ecke mit $r(x)$ und $r(v)$, sodass das Rechteck des zu x und v adjazenten Knotens korrekt platziert werden kann; wie in Abbildung 3.9 gezeigt. Die Invariante ist für $k + 1$ Rechtecke erfüllt.

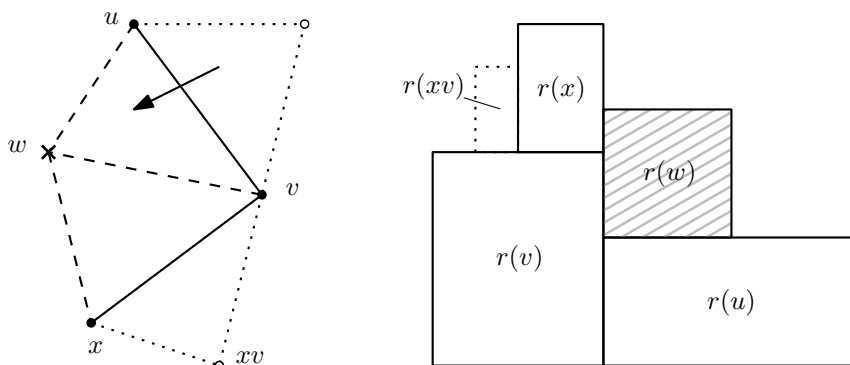


Abb. 3.9.: Freie Ecke beim Rechteck eines äußeren Dummy-Knotens

4. K-außenplanare Graphen

Möchte man den zu zeichnenden Graphen noch weiter verallgemeinern, ohne sich den planaren Graphen widmen zu müssen, so bieten sich die *k-außenplanaren* an. Diese bilden eine Teilmenge der planaren Graphen und enthalten die Klasse der einfach außenplanaren Graphen vollständig. Ein Graph $G = (V, E)$ ist genau dann *k-außenplanar*, wenn G ohne die Knoten und zugehörigen Kanten aus V^* genau $k - 1$ -außenplanar ist. Dabei sei $V^* \subseteq V$ die Menge der Knoten, die in G zur Außenfacette benachbart sind.

Prinzipiell wurden in Kapitel 3 schon Rechtecksdarstellungen nicht nur einfach außenplanarer Graphen gezeichnet, da die inneren Dummy-Knoten auch in die Darstellung eingefügt wurden. Dabei konnte jedoch die Größe der Fläche frei gewählt werden, was bei echten *k-außenplanaren* Graphen nicht mehr möglich sein wird.

Beschränkt man sich weiter auf 2-außenplanare Graphen, also jene, deren Graph einfach außenplanar ist, wenn man alle Knoten der Außenfacette entfernt, so existieren solche, deren Rechtecksdarstellung nicht gezeichnet werden kann. Man betrachte hierzu den Graphen aus Abbildung 4.1. Dieser ist ersichtlicher Weise 2-außenplanar. Dabei liegt das Rechteck $r(u_5)$ inmitten der kreisförmig verbundenen Rechtecke $r(u_1)$, $r(u_2)$, $r(u_3)$ und $r(u_4)$. Das Rechteck des Knotens u_0 sei zu diesen benachbart. Da man die zu u_5 benachbarten Rechtecke in der Darstellung ebenso kreisförmig um $r(u_5)$ anordnen muss, wie in Abbildung 4.1 zu sehen, kann $r(u_0)$ nicht mehr zu allen Vieren benachbart sein. Somit müsste der Graph zu Beginn oder während der Zeichnung darauf geprüft werden, ob eine Rechtecksdarstellung existiert.

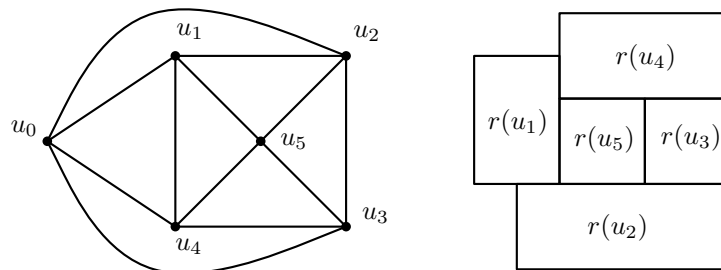


Abb. 4.1.: 2-außenplanarer Graph mit zugehöriger Rechtecksdarstellung ohne $r(u_0)$

5. Abschließende Worte und Ausblick

Ausgehend von maximal außenplanaren Graphen konnte in Kapitel 2 ein Algorithmus zur Zeichnung der entsprechenden Rechtecksdarstellung gezeigt werden. Dazu wurde die Eigenschaft der aus der Maximalität resultierenden Dreiecksfacetten ausgenutzt. Diese führen dazu, dass beim Betreten einer neuen Facette der Ort des zu zeichnenden Rechtecks eindeutig bestimmt ist. Im Übrigen geht auch van Kreveld [vKS07] im zweiten Kapitel seiner Arbeit über Rechteck-Kartogramme von triangulierten Rechtecksdarstellungen aus, also solche, deren Graph nur dreieckige Facetten im Inneren besitzt. Die Darstellung in Rechtecksform wurde daraufhin induktiv bewiesen, sowie ein Java Programm zur Umsetzung des Algorithmus vorgestellt.

In Kapitel 3 ist auf die Eigenschaft der dreieckigen Facetten im Ausgangsgraphen verzichtet worden. Die somit betrachteten außenplanaren Graphen wurden durch innere und äußere Dummy-Knoten wiederum in dreiecksförmige Facetten unterteilt, um ein effektives durchlaufen des Graphen zu ermöglichen. Auch dieses Verfahren konnte mithilfe von Induktion über die Anzahl der gezeichneten Rechtecke bewiesen werden.

Wagt man sich noch etwas weiter voran und betrachtet k -außenplanare Graphen, wie in Kapitel 4, so zeigt man einfach, dass schon bei 2-außenplanaren Graphen nicht mehr alle in Rechtecksdarstellung gezeichnet werden können.

Aufgrund der in Kapitel 1 beschriebenen Einschränkung auf die korrekte Darstellung der Nachbarschaftsbeziehungen, sowie die Fläche der Rechtecke, hat die Zeichnung nicht viel mit Grundrissdarstellungen zu tun. Diese setzen meist das Einhalten eines vorgegebenen Umrisses des Gebäudes voraus. Mehr Ähnlichkeit besteht zu Rechteckskartogrammen, wobei hier das Seitenverhältnis der Rechtecke eine wichtige Rolle spielt. Diese könnte man beispielsweise durch eine veränderte Aufteilung der freien Fläche oberhalb der Rechtecke berücksichtigen. Zudem hat das Seitenverhältnis der beiden zu Beginn gesetzten Rechtecke einen großen Einfluss auf die Seitenverhältnisse aller Rechtecke.

Das Durchschreiten des Graphen muss auch nicht unbedingt über eine zur Außenfacette benachbarte Kante erfolgen. Hier könnte die gemeinsame Ecke dreier Rechtecke innerhalb des Graphens gezeichnet werden und aufbauend darauf die umliegenden. Somit kann der gerade ausgewählte Bereich im richtigen Seitenverhältnis dargestellt werden, wobei sich die umliegenden Rechtecke diesem anpassen.

Ein Betrachten anderer Graphenklassen könnte ebenso zum Erfolg führen. Interessant sind beispielsweise serien-parallele Graphen, da diese rekursiv aufeinander aufbauen. Dabei können zwei Teilgraphen entweder in Serie, also hintereinander, oder parallel miteinander kombiniert werden.

Offensichtlich gibt es noch vieles über das man sich in Zusammenhang mit Rechtecksdarstellungen beschäftigen könnte, was jedoch den Rahmen dieser Bachelorarbeit sprengen würde.

Literaturverzeichnis

- [Che] Otfried Cheong: Ipe extensible drawing editor. <http://sourceforge.net/projects/ipe7/>.
- [Die00] Reinhard Diestel: *Graphentheorie*. Springer-Lehrbuch. Springer, 2. Auflage, 2000. <http://books.google.de/books?id=ivjuAAAAMAAJ>.
- [FR91] Thomas M. J. Fruchterman und Edward M. Reingold: Graph drawing by force-directed placement. *Software-Practice and Experience*, 21(11):1129–1164, 1991.
- [KN12] Sven Oliver Krumke und Hartmut Noltemeier: *Graphentheoretische Konzepte und Algorithmen*. Springer Vieweg, 3. Auflage, 2012.
- [KSK12] Reinhard Koenig, Sven Schneider und Katjha Knecht: *KREMLAS: Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau*. Verlag der Bauhaus-Universität Weimar, 2012.
- [Nav] Barak Naveh: JGraphT. <https://github.com/jgrapht/jgrapht>.
- [SvKF06] Bettina Speckmann, Marc van Kreveld und Sander Florisson: A Linear Programming Approach to Rectangular Cartograms. In: A. Riedl, W. Kainz und G.A. Elmes (Herausgeber): *Proceedings 12th International Symposium on Spatial Data Handling (SDH'06)*, Seiten 529–546. Springer, 2006. http://dx.doi.org/10.1007/3-540-35589-8_34.
- [vKS07] Marc van Kreveld und Bettina Speckmann: On rectangular cartograms. *Computational Geometry*, 37(3):175 – 187, 2007, ISSN 0925-7721. <http://www.sciencedirect.com/science/article/pii/S0925772106000770>, Special Issue on the 20th European Workshop on Computational Geometry (EW-CG'04).

Abbildungsverzeichnis

1.1. Verschiedene Arten der Raumaufteilung	4
1.2. System virtueller Federn	4
1.3. Die Bevölkerung Europas als Rechteck-Kartogramm [vKS07]	5
1.4. Lineare Begrenzungen des Rechtecks R [SvKF06]	6
2.1. Ein maximal außenplanarer Graph (durchgezogene Kanten), maximal planarer Graph (gestrichelte und durchgezogene Kanten) und sein Dualgraph (gepunktete Kanten)	8
2.2. Rechteck u	9
2.3. Setzen der ersten beiden Rechtecke	12
2.4. G_5 mit fünf gesetzten Rechtecken	13
2.5. u_p und u_q vor dem Setzen von u_z	13
2.6. u_p und u_q nach dem Setzen von u_z	13
2.7. Weitere Möglichkeiten Knoten zu G' hinzuzufügen, nachdem u_z gesetzt wurde	14
2.8. Der als Rechtecksdarstellung zu zeichnende Graph	14
2.9. Die Rechtecksdarstellung als Ausgabe des Java Programms mit der Beschriftung [Name, Größe]	15
3.1. Auswirkung des Entfernens einer Kante auf die Rechtecksdarstellung	16
3.2. Der außenplanare Graph G (durchgezogene Linien und Punkte) und seine Erweiterung G' (gestrichelte Linien und Kreuze)	17
3.3. Bestimmung des freien Bereichs als erster Schritt beim Durchlaufen einer λ -eck-Facette	19
3.4. Die Rechtecke werden auf der freien Fläche bündig platziert	19
3.5. Fertige Rechtecksdarstellung einer λ -eck-Facette	20
3.6. Der äußere Dummy-Knoten kann wie ein normaler Knoten behandelt werden	20
3.7. Existenz von zur Außenfacette benachbarter Kanten $e \in E$	21
3.8. Das λ -Eck umgebende Rechtecke und ihre Platzierung in der Rechtecksdarstellung	23
3.9. Freie Ecke beim Rechteck eines äußeren Dummy-Knotens	23
4.1. 2-außenplanarer Graph mit zugehöriger Rechtecksdarstellung ohne $r(u_0)$	24

Liste der Algorithmen

1.	MAXOUTERPLANARRECTANGULAR(G)	10
2.	INITIALIZE(G)	10
3.	START(G, G^*, Q)	11
4.	PLACE(u, v, w)	11

A. Dateiliste der beigelegten CD

CDinhalt.txt

/Arbeit:

Rechtecksdarstellung gewichteter außenplanarer Graphen.pdf

/Java:

./Java:

jgraph-0.9.0.tar.gz

maxplanar

./Java/maxplanar:

DrawJGraph.java

FaceNode.java

Graph.java

IpeDraw.java

IpeDrawTest.java

MaxOuterPlanarToRectangle.java

Point.java

Rectangle.java

RoomNode.java

/Quellen:

Kremlas.pdf

A Linear Programming Approach to Rectangular Cartograms.pdf

Graphentheoretische Konzepte und Algorithmen.pdf

On rectangular cartograms.pdf

B. Eidesstattliche Erklärung

Ich, Lorenz Reinhart, Matrikel-Nr. 1807540, versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema

Rechtecksdarstellung gewichteter außenplanarer Graphen

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Würzburg, den 5. September 2014

LORENZ REINHART