

FernUniversität in Hagen  
Fakultät für Mathematik und Informatik  
Lehrgebiet für Theoretische Informatik  
Prof. Dr. André Schulz

# Graphzeichnen mit niedriger visueller Komplexität auf dem Gitter

Diplomarbeit im Studiengang Informatik

Dr. med. dent. Gregor Hültenschmidt  
Matrikelnummer 2523140

14.10.2016

betreut von:

Dr. Philipp Kindermann  
Lehrgebiet Theoretische Informatik  
FernUniversität in Hagen

# Kurzzusammenfassung

Von zentraler Bedeutung bei der Visualisierung von Graphen ist die Fragestellung, welche Faktoren die Lesbarkeit einer Graphenzeichnung beeinflussen. Ein positiver Einflussfaktor könnte eine niedrige visuelle Komplexität der Zeichnung sein. Visuelle Komplexität wird definiert als Kardinalität der Grundmenge von geometrischen Objekten, mit denen die Kanten des Graphen gezeichnet werden. Sie kann zum Beispiel vermindert werden, indem mit einem einzigen geraden Liniensegment mehr als eine Kante gezeichnet wird und das Segment somit innere Knoten hat. Im Folgenden sei mit  $n$  die Anzahl der Knoten und mit  $e$  die Anzahl der Kanten eines Graphen bezeichnet. Im Rahmen der vorliegenden Arbeit wurden für verschiedene Graphenklassen Algorithmen entwickelt, die Zeichnungen auf dem Gitter mit geradlinigen Segmenten und niedriger visueller Komplexität erstellen. Darüber hinaus ist die von einer solchen Zeichnung benötigte Fläche eine Polynomialfunktion von  $n$ . Im Einzelnen wurden folgende visuelle Komplexität  $s$  und Gittergröße  $\Delta$  erreicht. 1. Für Bäume  $s = \lceil 3/4 e \rceil$  und  $\Delta = O(n^2) \times O(n^{1,58})$ . 2. Für 3-Bäume  $s = 8/9 e + O(1)$  und  $\Delta = O(n) \times O(n^2)$ . 3. Für maximale außenplanare Graphen  $s = 3/4 e + O(1)$  und  $\Delta = O(n) \times O(n^2)$ . 4. Für serien-parallele Graphen  $s = 2\xi - 1$ , wobei  $\xi$  die Anzahl der Ohren in einer verschachtelten offenen Ohrenzerlegung des Graphen ist, und  $\Delta = O(n) \times O(n^2)$ . 5. Für serien-parallele Graphen mit Grad drei eine nahezu minimale visuelle Komplexität und  $\Delta = O(n) \times O(n^2)$ . 6. Für vollständig balancierte 3-Bäume eine minimale visuelle Komplexität und  $\Delta = O(n) \times O(n)$ . Alle Algorithmen wurden prototypisch implementiert. Das für Bäume erzielte Ergebnis fand Eingang in [25]. Hinsichtlich des Zusammenhangs zwischen Lesbarkeit und visueller Komplexität einer Graphenzeichnung wird eine Experimentalstudie angeregt.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>VI</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Problemstellung . . . . .	1
1.2. Allgemeine Definitionen . . . . .	2
1.3. Bisherige Erkenntnisse . . . . .	4
1.4. Konkretisierung der Themenstellung . . . . .	6
<b>2. Gitterzeichnungen mit niedriger visueller Komplexität</b>	<b>7</b>
2.1. Segmentreduzierte Gitterzeichnungen von Bäumen . . . . .	7
2.1.1. Spezifische Grundlagen . . . . .	7
2.1.2. Algorithmus TREEFEWSEG . . . . .	8
2.1.3. Zusammenfassung . . . . .	17
2.2. Segmentreduzierte Gitterzeichnungen von 3-Bäumen . . . . .	17
2.2.1. Spezifische Grundlagen . . . . .	17
2.2.2. Beschreibung von Algorithmus 3TREEFEWSEG . . . . .	28
2.2.3. Eigenschaften von Algorithmus 3TREEFEWSEG . . . . .	30
2.2.4. Beweis von Algorithmus 3TREEFEWSEG: . . . . .	32
2.2.5. Zentrales Ergebnis . . . . .	48
2.3. Segmentminimale Gitterzeichnungen von vollständig balancierten 3-Bäumen . . . . .	55
2.3.1. Spezifische Grundlagen . . . . .	55
2.3.2. Algorithmus 3TREEOPTSEG . . . . .	57
2.3.3. Algorithmus BALANCED3TREEOPTSEG . . . . .	60
2.4. Segmentreduzierte Gitterzeichnungen von maximalen außenplanaren Graphen . . . . .	61
2.4.1. Spezifische Grundlagen . . . . .	61
2.4.2. Algorithmus OUTERPLANARFEWSEG . . . . .	66
2.5. Segmentreduzierte Gitterzeichnungen von serien-parallelen Graphen . . . . .	74
2.5.1. Spezifische Grundlagen . . . . .	74
2.5.2. Beschreibung von Algorithmus SPFEWSEG . . . . .	77
2.5.3. Ohrenzerlegung durch Algorithmus FEASIBLEDECOMPOSITION . . . . .	85
2.5.4. Folgerungen für Algorithmus SPFEWSEG . . . . .	90
2.5.5. Algorithmus SPDEG3FEWSEG . . . . .	97
2.5.6. Algorithmus SPFEWSEGUNIV . . . . .	104
<b>3. Implementierung</b>	<b>106</b>
3.1. Quellcode . . . . .	106
3.2. Systemvoraussetzungen und Programmstart . . . . .	106
3.3. Beschreibung des Programms . . . . .	107
<b>4. Zusammenfassung und Ausblick</b>	<b>110</b>
<b>A. Inhalt der beigefügten CD</b>	<b>113</b>
<b>Literatur</b>	<b>114</b>

# Abbildungsverzeichnis

1.	Die schweren Pfade eines Baums, in Anlehnung an [41]. . . . .	8
2.	Eine sichere Box und ihre Abmessungen. . . . .	9
3.	Zeichnung eines schweren Pfads. . . . .	10
4.	Eine Paarbox und ihre Abmessungen. . . . .	10
5.	Anordnung der sicheren Boxen in der ersten Runde. . . . .	11
6.	Eine neu konstruierte Box $B_{\text{neu}}$ und ihre Abmessungen, 1.Fall . . . . .	12
7.	Eine neu konstruierte Box $B_{\text{neu}}$ und ihre Abmessungen, 2.Fall . . . . .	13
8.	Abmessungen von $B_{\text{neu}}$ bei ungerader Anzahl der Boxen. . . . .	15
9.	Schnyder-Realisierer, in Anlehnung an [6]. . . . .	19
10.	Schnyder-Realisierer mit Nachfolgefunktion, in Anlehnung an [6]. . . . .	20
11.	Schnyder-Realisierer für den $K_4$ . . . . .	20
12.	Schnyder-Realisierer eines 3-Baums. . . . .	21
13.	Erster Teil im Beweis von Hilfssatz 6. . . . .	22
14.	Ausgeschlossene Lage einer Sehne bezüglich der Kontur. . . . .	23
15.	Existenz der Kante zwischen ganz linkem und ganz rechtem Vorfahren. . . .	24
16.	Erste und letzte horizontale Kante in einem Schnyder-Realisierer. . . . .	25
17.	Darstellung und Beweis von Hilfssatz 10, nach [14]. . . . .	26
18.	Verschiebung von $v_2$ in Richtung der Grundlinie. . . . .	27
19.	Darstellung des Beweises von Hilfssatz 11. . . . .	28
20.	Darstellung des Beweises zu Hilfssatz 12. . . . .	28
21.	OpDiagShift im Überblick. . . . .	30
22.	Verwendete Definitionen in Algorithmus 3TREEFEWSEG. . . . .	32
23.	Erster Fall: Domäne von $v_{k+1}$ und $v_r$ mit $(v_\ell, v_r) \in T_1$ . . . . .	34
24.	Zweiter Fall: Domäne von $v_{k+1}$ und $v_r$ mit $(v_\ell, v_r) \in T_2$ . . . . .	35
25.	Verschiebung von $(v_{k+1}, v_r)$ im Fall $v_i = v_{k+1}$ . . . . .	36
26.	Verschiebung von $(v_i, v_j)$ im Fall $v_i \in C_{k+1}^*$ . . . . .	37
27.	Veränderung der Steigung der $T_n$ -Kanten während OpDiagShift. . . . .	39
28.	Verschiebung der letzten horizontalen Kante. . . . .	40
29.	Letzte horizontale Kante mit ihrer korrespondierenden $T_1$ -Kante. . . . .	41
30.	OpInsert im Detail, Fall (a). . . . .	42
31.	OpInsert im Detail, Fall (b.1). . . . .	43
32.	OpInsert im Detail, Fall (b.2). . . . .	43
33.	OpDiagShift im Detail, Teil 1. . . . .	44
34.	OpDiagShift im Detail, Teil 2. . . . .	45
35.	Verschiebung der $T_n$ -Kanten. . . . .	46
36.	Der zu zeichnende 3-Baum. . . . .	50
37.	Zeichnungen $\Gamma_3$ bis $\Gamma_9$ . . . . .	51
38.	Zeichnung $\Gamma_{10}$ , zusätzlich mit möglicher Optimierung. . . . .	52
39.	Zeichnung $\Gamma_{11}$ ohne vorangegangene Optimierung. . . . .	53
40.	Zeichnung $\Gamma_{11}$ mit vorangegangener Optimierung. . . . .	54

41.	Repräsentativer Knoten $p$ mit verschachtelten Dreiecken, nach [33]. . . . .	55
42.	Ein 3-Baum und sein repräsentativer Baum, nach [33]. . . . .	56
43.	Baryzentrische Koordinaten. . . . .	56
44.	Repräsentativer Baum und zugehörige Zeichnung $\Gamma_3$ . . . . .	57
45.	Kollinearität von $U, P, S$ . . . . .	59
46.	Kanonische Ordnung für einen maximal außenplanaren Graphen. . . . .	62
47.	Zwei verschiedene Einbettungen eines maximalen außenplanaren Graphen. .	63
48.	Zerlegung in die beiden Bäume $T_1$ und $T_2$ . . . . .	64
49.	Bildung eines neuen Blatts in den beiden Bäumen. . . . .	65
50.	Nächster nicht verwandter Knoten. . . . .	66
51.	Beweis von Hilfssatz 17. . . . .	66
52.	Steigungslabls. . . . .	68
53.	Gezeichneter Referenzbaum $T_1$ in verschiedenen Stadien. . . . .	68
54.	Maximaler außenplanarer Graph $G$ und segmentreduzierte Zeichnung. . .	69
55.	Kreuzungsfreiheit der Kanten des Referenzbaums. . . . .	71
56.	Kreuzungsfreiheit der Kanten des komplementären Baums. . . . .	72
57.	Untere Schranke für die Gittergröße. . . . .	73
58.	Verschachtelte offene Ohrenzerlegung und Spannbaum. . . . .	77
59.	Terminologie der Verschachtelungsstruktur. . . . .	78
60.	Ansprechende und beliebige verschachtelte offene Ohrenzerlegung. . . . .	78
61.	Triviale Ohren und ihre Modifikation. . . . .	79
62.	Belegung der Labels und Stacks. . . . .	80
63.	Zeichenfolge mit reduzierter Segmentzahl für serien-parallelen Graphen. .	81
64.	Offener Sektor eines Knotens. . . . .	82
65.	Letzte Kanten, die zu einem gemeinsamen Endpunkt inzident sind. . . . .	83
66.	Definition der Domäne eines Knotens. . . . .	84
67.	Kreuzungsfreiheit der letzten Kanten. . . . .	85
68.	Reihenfolge während der Konstruktion der Ohren. . . . .	86
69.	Korrektheit von Algorithmus FEASIBLEDECOMPOSITION. . . . .	88
70.	Krümmung eines Ohrs gegen den Uhrzeigersinn. . . . .	91
71.	Serien-paralleler Graph und kombinatorisch isomorpher Graph. . . . .	91
72.	Verschachtelte Ohren mit Krümmung gegen den Uhrzeigersinn. . . . .	94
73.	Zeichnung eines 2-Baums mit Algorithmus SPFEWSEG. . . . .	96
74.	Graph mit $E_{\text{inner}}$ und nach Entfernen von $E_{\text{inner}}$ . . . . .	97
75.	Gerüst vom Typ I und Typ II. . . . .	98
76.	Prozessieren der Gerüste, Teil 1. . . . .	100
77.	Prozessieren der Gerüste, Teil 2. . . . .	100
78.	Prozessieren der Gerüste, Teil 3. . . . .	101
79.	Zeichnung eines serien-parallelen Graphen mit Algorithmus SPDEG3FEWSEG.	102
80.	Illustration der Verringerung der Segmentanzahl. . . . .	102
81.	Quelle und Senke bei nicht 2-zusammenhängendem serien-parallelen Graphen.	105

# Tabellenverzeichnis

1.	Obere Schranke für die Segmentanzahl bei verschiedenen Graphenklassen. . .	5
2.	Visuelle Komplexität bei Zeichnungen mit Kreisbögen, nach [41]. . . . .	6
3.	Inhalt der beigefügten CD. . . . .	113

# 1. Einleitung

Zunächst wird die Problemstellung beschrieben, insbesondere der Zusammenhang zwischen Lesbarkeit eines Graphen und visueller Komplexität. Nach einem Überblick über die in diesem Bereich verwendeten Definitionen werden die hinsichtlich der Fragestellung relevanten bisherigen Forschungsergebnisse aufgeführt. In Kenntnis des aktuellen Forschungsstandes werden dann Themenstellung und Zielsetzung konkretisiert.

## 1.1. Problemstellung

Der Informationsgehalt eines Graphen wird in der Regel nicht durch seine mathematische Notation, sondern mit Hilfe einer Visualisierung vermittelt. Typische Anwendungen sind die Modellierung sozialer Netzwerke, Darstellung von Produktionsabläufen, Organigramme und Visualisierung von Geodaten. Die Knoten eines Graphen repräsentieren hierbei Entitäten der Anwendungsdomäne, während die Kanten Beziehungen zwischen den Entitäten modellieren.

Informal besteht ein Graph aus einer Knoten- und einer Kantenmenge, wobei eine einzelne Kante zwei Knoten als ihre Endknoten aufweist. Häufig werden die Knoten des Graphen durch Punkte in der Zeichenebene dargestellt. Eine Kante wird dann mit einer Verbindungslinie zwischen den beiden Punkten gezeichnet, die ihren Endknoten entsprechen. Die konkrete Form dieser Linien richtet sich dabei nach dem gewählten Zeichenstil. Denkbar sind Kreisbögen, geschwungene, geknickte oder gerade Linien. In der Regel soll eine Zeichnung gewisse Ästhetikkriterien erfüllen. Solche Ästhetikkriterien sind zum Beispiel eine möglichst geringe Fläche der Zeichnung, möglichst große Winkelauflösung, möglichst geringe Anzahl von sich kreuzenden Kanten, harmonisches Verhältnis von Breite zu Höhe der Zeichnung, eine symmetrische Darstellung des Graphen und ein möglichst großes Längenverhältnis zwischen kürzester und längster Kante. Nicht alle dieser Kriterien lassen sich gleichzeitig verwirklichen. Die Priorität wird je nach Anwendung auf unterschiedliche Kriterien gelegt. So ist zum Beispiel im Chipdesign die Fläche der Zeichnung maßgebend. Eine Verbesserung hinsichtlich eines Kriteriums kann zu einer Verschlechterung hinsichtlich eines anderen Kriteriums führen. In einem solchen Fall muss dann abhängig vom Kontext zwischen den einzelnen Kriterien abgewogen werden. Die Anwendbarkeit eines Zeichenstils oder Ästhetikkriteriums kann von der Struktur des betrachteten Graphen abhängig sein.

Zeichenstil und Ästhetikkriterien haben Einfluss auf die Lesbarkeit eines Graphen. Durch kognitive Experimentalstudien wurde nachgewiesen, dass insbesondere die Minimierung der Anzahl von Kantenkreuzungen die Lesbarkeit der Graphenzeichnung verbessert [36]. Idealerweise sollte eine Zeichnung gar keine sich kreuzenden Kanten aufweisen. Unlängst wurde die Frage gestellt, ob die menschliche Wahrnehmung eines Graphen durch die Anzahl der geometrischen Objekte beeinflusst wird, in die ein Graph zerlegt wird [41]. Die Kardinalität der Grundmenge von Objekten, aus denen sich der Graph zusammensetzt, wird hierbei als visuelle Komplexität bezeichnet. Ein einzelnes geometrisches Objekt ist zum Beispiel ein Knoten, ein Kreisbogen zwischen zwei Knoten oder eine gerade Strecke zwischen zwei Knoten. Als ein einziges Objekt zählt aber auch ein einzelnes gerades Segment, mit

dem mehrere Kanten des Graphen so gezeichnet werden, dass alle Endknoten dieser Kanten auf dem betrachteten einzelnen Segment liegen. So kann mit einem einzelnen Segment mehr als eine Kante gezeichnet werden.

Übergeordnet besteht somit die Fragestellung, ob eine geringere visuelle Komplexität die Lesbarkeit eines Graphen verbessert. Für die Beantwortung dieser Frage ist es notwendig, Algorithmen zu entwickeln, die einen gegebenen Graphen mit einer geringeren visuellen Komplexität zeichnen als herkömmlich.

## 1.2. Allgemeine Definitionen

Es werden die grundlegenden Definitionen aus der allgemeinen Graphentheorie zusammengestellt, die im weiteren Verlauf benötigt werden. Die Darstellung ist teilweise an [11] angelehnt. Formal wird ein Graph  $G$  als ein Paar  $G = (V, E)$  zweier Mengen  $V$  und  $E$  mit  $E \subseteq V \times V$  beschrieben. Dabei ist  $V$  die Menge der Knoten von  $G$ , während  $E$  die Menge der Kanten von  $G$  ist. In einer Zeichnung von  $G$  wird jedem Knoten  $v \in V$  ein unterschiedlicher Punkt der Ebene zugewiesen. Jede Kante  $(s, t) \in E$  wird auf eine zusammenhängende Kurve abgebildet, die in  $s$  beginnt und in  $t$  endet. Kein weiterer Knoten des Graphen darf auf dieser Kurve gezeichnet werden, da die Zeichnung sonst nicht eindeutig wäre. Die Form der Kurve ist zunächst nicht festgelegt. Je nach Zeichenstil kann diese Kurve eine gerade Strecke, ein Kreisbogen oder eine geschwungene Kurve sein. Eine Kante und ein Endknoten von ihr sind zueinander inzident. Es wird nicht immer streng unterschieden zwischen einem Knoten und seiner Zeichnung als Punkt beziehungsweise zwischen einer Kante und ihrer Zeichnung als Kurve. Wenn je zwei solcher Kurven außer eventuell gemeinsamen Endpunkten keine weiteren gemeinsamen Punkte haben, nennt man die Zeichnung eben. Man sagt in einem solchen Fall auch, dass die Kanten kreuzungsfrei gezeichnet werden. Ein Graph, für den eine ebene Zeichnung angefertigt werden kann, heißt planar. Ein maximaler planarer Graph ist ein Graph, der nach Hinzufügen einer beliebigen Kante nicht mehr planar ist. Jeder planare Graph kann durch Triangulierung zu einem maximalen planaren Graph erweitert werden. Einen maximalen planaren Graphen nennt man auch einen ebenen Dreiecksgraphen. Unter der Einbettung eines Graphen versteht man die Auflistung (im Uhrzeigersinn) der Kanten um jeden Knoten herum. Durch die Knoten und Kanten entstehen in der Zeichnung des Graphen zusammenhängende Gebiete. Die Außenfläche ist das einzige Gebiet, das unbeschränkt und somit gegenüber den anderen Gebieten ausgezeichnet ist. Die Außenfläche eines ebenen Dreiecksgraphen ist ein Dreieck. Ein ebener Dreiecksgraph ist durch seine Einbettung und seine Außenfläche eindeutig bestimmt. Eine Zeichnung eines Graphen, in der die Knoten ganzzahlige Koordinaten haben, nennt man eine Gitterzeichnung. Als Gitterschranke wird in der vorliegenden Arbeit die Höhe und Breite der Zeichnung in Abhängigkeit von der Knotenanzahl  $n$  bezeichnet. Eine polynomiale Gitterschranke liegt vor, wenn in der asymptotischen Notation Höhe und Breite durch ein Polynom in  $n$  beschrieben werden können. Man sagt dann auch, dass ein Graph auf einem polynomialen Gitter gezeichnet wird. Bei einer exponentialen Gitterschranke



steigt die Fläche der Zeichnung exponentiell mit  $n$  an. Nur Gitterzeichnungen auf einem polynomialen Gitter sind von praktischem Nutzen.

Ein Weg, auch Pfad genannt, ist eine Abfolge von Knoten, in der jeder Knoten mit seinem Vorgänger und Nachfolger durch eine Kante verbunden ist. Der Begriff Kantenzug bezeichnet dasselbe, legt jedoch die Betonung mehr auf die Kanten zwischen den Knoten. Wenn Start- und Endknoten identisch sind, bezeichnet man den Weg als geschlossen oder auch als Kreis. Ein Hamiltonkreis enthält jeden Knoten des Graphen genau einmal. Wenn den Kanten eines Graphen eine Richtung zugeordnet wird, wird der Graph gerichtet genannt, andernfalls ungerichtet. Ein Graph wird vollständig genannt, wenn jeder seiner Knoten mit allen anderen Knoten durch eine Kante verbunden ist. Der vollständige Graph mit  $n$  Knoten wird mit  $K_n$  bezeichnet. Ein (ungerichteter) Graph ist zusammenhängend, wenn es zwischen je zwei Knoten einen Weg im Graphen gibt. Einen Graphen nennt man  $k$ -knotenzusammenhängend, wenn auch nach Entfernen von beliebigen  $(k-1)$  Knoten der so entstandene Graph noch zusammenhängend ist. Verwendet man als Kriterium das Entfernen von Kanten, erhält man den entsprechenden Begriff des  $k$ -Kantenzusammenhangs. Der Begriff  $k$ -zusammenhängend wird als Synonym für  $k$ -knotenzusammenhängend verwendet. Einen Knoten eines zusammenhängenden Graphen bezeichnet man als Artikulationsknoten, wenn nach Entfernen dieses Knotens der entstandene Graph nicht mehr zusammenhängend ist. Für einen Graphen  $G = (V, E)$  und  $V' \subseteq V$  beinhaltet der von  $V'$  aufgespannte Untergraph  $G'$  von  $G$  alle Kanten aus  $E$ , deren beide Endknoten Element von  $V'$  sind. Wenn ein Untergraph  $G' = (V', E')$  vollständig ist, nennt man  $V'$  eine  $k$ -Clique, wobei  $k = |V'|$  ist. Ein Graph  $G'' = (V'', E'')$  ist ein Teilgraph von  $G$ , wenn  $V'' \subseteq V$  und  $E'' \subseteq E$  ist. Ein Teilgraph  $T = (V_T, E_T)$  von  $G = (V_G, E_G)$  heißt Spannbaum von  $G$ , wenn  $T$  ein Baum ist und  $V_T = V_G$  gilt. Die Nachbarn eines Knotens sind all die Knoten, die mit ihm über eine Kante verbunden sind. Der Grad eines Knotens ist die Anzahl seiner Nachbarn. Der Grad eines Graphen ist der maximale Grad, den ein Knoten des Graphen hat. Ordnet man für jeden Knoten seine Nachbarn in einer Liste an, erhält man die Adjazenzlistendarstellung des Graphen. Eine Schlinge ist eine Kante, deren beide Endknoten identisch sind. Zwei Kanten sind Multikanten, wenn ihre Endknoten wechselseitig identisch sind.

Je nach Eigenschaften werden Graphen in Graphenklassen eingeteilt. Bei den planaren Graphen sind Bäume, 2-Bäume, 3-Bäume, serien-parallele Graphen, außenplanare Graphen und planare kubische Graphen wichtige Unterklassen. Ein Baum ist ein zusammenhängender Graph, der keinen Kreis hat. Ein 2-Baum entsteht, wenn ausgehend von einem Dreieck wiederholt ein Knoten hinzugefügt und mit zwei benachbarten Knoten verbunden wird. Bei einem 3-Baum wird ein Dreieck rekursiv in drei kleinere Dreiecke aufgeteilt. Ein serien-paralleler Graph entsteht durch rekursive Anwendung von seriellen und parallelen Verkettungen. Bei einem außenplanaren Graphen liegt jeder Knoten auf der Außenfläche. In einem kubischen Graphen hat jeder Knoten den Grad drei. Die formalen Definitionen folgen in den einzelnen Unterkapiteln.

Häufig zeichnet man in einem Baum einen Knoten besonders aus und nennt ihn Wurzel. Durch die Wahl einer Wurzel kann man die Kanten des Baums orientieren. In der Regel

orientiert man die Kanten so, dass sie weg von der Wurzel weisen. In diesem Fall hat jeder Knoten mit Ausnahme der Wurzel genau eine eingehende Kante. Dann bilden diejenigen Knoten, die von einem Knoten  $v$  über eine von  $v$  ausgehende Kante erreicht werden können, den an  $v$  verwurzelten Unterbaum. Den Durchlauf durch einen Baum, in dem rekursiv zunächst ein Knoten und dann der an ihm verwurzelte Unterbaum besucht wird, nennt man einen Preorder-Durchlauf. Den Durchlauf durch einen Baum, in dem rekursiv zunächst der an einem Knoten verwurzelte Unterbaum und dann der Knoten selbst besucht wird, nennt man einen Postorder-Durchlauf. Orientiert man die Kanten zur Wurzel hin, müssen in den Definitionen die beiden Begriffe eingehend und ausgehend vertauscht werden.

### 1.3. Bisherige Erkenntnisse

In Existenzbeweisen zeigten Wagner und Fáry unabhängig voneinander, dass jeder planare Graph so visualisiert werden kann, dass die Kanten mit kreuzungsfreien, geraden Strecken gezeichnet werden [44][16]. Für 3-zusammenhängende Graphen gab Tutte einen Berechnungsalgorithmus für solche Zeichnungen an [43]. Dieser Algorithmus führt jedoch zu rationalen Koordinaten. Fordert man als Nebenbedingung ganzzahlige Koordinaten, führt er daher zu einer exponentialen Gitterschranke der entstehenden Gitterzeichnung. Es blieb lange ungeklärt, ob für ebene Dreiecksgraphen eine ebene Gitterzeichnung mit geraden Strecken angefertigt werden kann, deren Fläche asymptotisch quadratisch zur Anzahl  $n$  der Knoten ist. In [20] wurde gezeigt, dass eine solche Zeichnung eines ebenen Dreiecksgraphen mit kreuzungsfreien, geraden Strecken auf einem  $(2n-4) \times (n-2)$  Gitter erfolgen kann. Diese Methode wird nach der Art des zugrunde liegenden Algorithmus auch als Shift-Methode bezeichnet. Sie verwendet eine kanonische Ordnung der Knoten. Das Ordnungsprinzip ist so gewählt, dass die Knoten gemäß der Ordnung sukzessive auf der Außenfläche hinzugefügt werden können. Unabhängig davon wurde in [40] gezeigt, wie ein ebener Dreiecksgraph ebenfalls nur mit kreuzungsfreien, geraden Strecken unter Verwendung eines sogenannten Schnyder-Realisierers auf einem  $(n-2) \times (n-2)$  großen Gitter gezeichnet werden kann. Ein Schnyder-Realisierer ist vereinfacht dargestellt eine Einteilung der Kantenmenge in drei Partitionen, so dass jede Partition einen an einem Knoten der Außenfläche verwurzelten Baum bildet. Von beiden Algorithmen existieren Varianten und Verfeinerungen. So gibt es einen Algorithmus, der auch mit der Shift-Methode ein  $(n-2) \times (n-2)$  großes Gitter erzielt [8]. Durch Triangulierung kann jeder planare Graph zu einem ebenen Dreiecksgraphen erweitert werden. Dieser Dreiecksgraph kann dann mit der Shift- oder Realisierer-Methode auf einem Gitter der Größe  $(n-2) \times (n-2)$  gezeichnet werden. Nach Entfernen der durch die Triangulierung hinzugefügten Kanten liegt eine Zeichnung des ursprünglichen Graphen vor. Jeder planare Graph kann daher auf einem  $O(n) \times O(n)$  großen Gitter kreuzungsfrei mit geraden Strecken als Kanten gezeichnet werden.

Algorithmen zur Verringerung der visuellen Komplexität wurden erstmals in [12] vorgestellt. Ein Segment wird dort definiert als eine maximale Menge von Kanten, die eine gerade Strecke bilden. Für verschiedene Graphenklassen konnte die für die Zeichnung eines

Graphen minimale Segmentanzahl—oder eine obere Schranke hierfür—in Abhängigkeit von der Knotenzahl  $n$  bestimmt werden. Dies erfolgte für Bäume, 2-Bäume, 3-Bäume, maximale außenplanare Graphen und 3-zusammenhängenden Graphen. In der Folgezeit sind Algorithmen für weitere Graphenklassen hinzugekommen. Die einzelnen Schranken können der Tabelle 1 entnommen werden. Sofern möglich wurde für die einzelnen Graphenklassen die Kantenanzahl  $e$  in Abhängigkeit von der Knotenzahl  $n$  aufgeführt. Dadurch können die Kantenanzahl eines Graphen und die minimale Anzahl der geraden Liniensegmente in einer Zeichnung des Graphen miteinander verglichen werden.

**Tabelle 1:** Obere Schranke für die Segmentanzahl im Vergleich zur Kantenanzahl bei verschiedenen Graphenklassen. Mit  $\eta$  sei die Anzahl der Knoten mit ungeradem Grad bezeichnet.

Graphenklasse	Kantenanzahl	Segmente	optimal	Referenz
Bäume	$n - 1$	$1/2 \eta$	ja	[12]
2-Bäume	$2n - 3$	$3/2 n$	ja	[12]
3-Bäume	$3n - 6$	$2n - 2$	ja	[12]
maximal außenplanar	$2n - 3$	$n$	ja	[12]
3-zusammenhängend	$e$	$5/2 n - 3$	nein	[12]
kubisch 3-zusammenhängend	$3/2 n$	$1/2 n - 3$	ja	[26][32]
ebener Dreiecksgraph	$3n - 6$	$7/3 n - \text{const}$	nein	[14]

Das gemeinsame Prinzip der Algorithmen lässt sich so beschreiben: Zuerst wird eine der Struktur des Graphen entsprechende Dekomposition des Graphen durchgeführt. Anschließend werden sukzessive einzelne Knoten oder Knotenmengen der wachsenden Zeichnung so hinzugefügt, dass der gewünschte Effekt der Reduzierung der Segmentanzahl erreicht wird. Alle aufgeführten Algorithmen erstellen jedoch keine Gitterzeichnung. Einzelne Segmente können daher beliebig klein werden. Bei der algorithmischen Berechnung können die bekannten Probleme der Gleitkommaarithmetik wie Rundungsfehlerausbreitung auftreten.

In ästhetisch ansprechenden Zeichnungen insbesondere von sozialen Netzwerken wurden in jüngerer Zeit neben geraden Strecken auch Kreisbögen zur Darstellung von Beziehungen verwendet [23]. Für die formale Betrachtung solcher Graphen lassen sich gerade Strecken als entartete Kreisbögen interpretieren. Bäume mit Kreisbögen wurden in [13] im Hinblick auf Winkelauflösung untersucht. Es ließ sich eine optimale Winkelauflösung erzielen. Diese Zeichnungen waren jedoch keine Gitterzeichnungen. Algorithmen zum Erstellen von Zeichnungen mit Kreisbögen mit niedriger visueller Komplexität finden sich in [41]. Die visuelle Komplexität wird dort nicht in Abhängigkeit von der Knotenzahl beschrieben, sondern als Verhältnis der Anzahl der Kreisbögen in der Zeichnung zur Anzahl der Kanten im Graphen. Tabelle 2 gibt für verschiedene Graphenklassen einen Überblick über die erzielten Resultate.

Der Algorithmus aus [41] zum Zeichnen von Bäumen mit Kreisbögen soll noch näher betrachtet werden. Dieser Algorithmus erstellt Gitterzeichnungen mit niedriger visueller Komplexität. Er basiert auf dem Konzept der sicheren Box. Nach Dekomposition des Baums in disjunkte Pfade wird für jeden Pfad der Knoten bestimmt, der am nächsten zur Wurzel liegt. Einem Pfad wird dann der an diesem Knoten verwurzelte Unterbaum zugeordnet.

**Tabelle 2:** Obere Schranken für die visuelle Komplexität von Zeichnungen mit Kreisbögen für verschiedene Graphenklassen. Mit  $e$  ist die Anzahl der Kanten des jeweiligen Graphen bezeichnet. Tabelle in Anlehnung an [41].

Graphenklasse	Visuelle Komplexität	Gitter	Referenz
Bäume	$3/4 e$	$O(n^{1.81}) \times n$	[41]
serien-parallel	$1/2 e + 1$	nein	[41]
3-Bäume	$11/18 e + 3$	nein	[41]
3-zusammenhängend	$2/3 e$	nein	[41]

Die in dem betrachteten Unterbaum gelegenen Knoten werden nun in einer sicheren Box positioniert. Mit einer Bottom-up-Strategie werden die sicheren Boxen sukzessive aus zuvor konstruierten Boxen erzeugt, bis der gesamte Baum gezeichnet ist. Die dabei erzielte Gitterschranke ist polynomial. Außer diesem Algorithmus ist lediglich ein weiterer Algorithmus bekannt, der Gitterzeichnungen mit niedriger visueller Komplexität auf einem polynomialen Gitter erstellt. Der in [32] und [26] beschriebene Algorithmus fertigt für planare kubische 3-zusammenhängende Graphen Gitterzeichnungen mit nahezu minimaler visueller Komplexität auf einem Gitter der Größe  $O(n) \times O(n)$  an.

Gitterzeichnungen auf einem polynomialen Gitter haben den Vorteil, dass aufgrund des Gitters die Kanten nicht beliebig klein werden können und die erzielten Zeichnungen wegen der polynomialen Gitterschranke nutzbar bleiben. In eventuellen Studien können sie daher gut verwendet werden, um den Einfluss der visuellen Komplexität auf die Lesbarkeit von Graphen zu untersuchen. Auch können diese Zeichnungen numerisch robust mit Ganzzahlarithmetik berechnet werden.

## 1.4. Konkretisierung der Themenstellung

Wie erwähnt werden in [12] und [14] Algorithmen für die Anfertigung von Zeichnungen mit wenigen Segmenten beschrieben. Hierbei stellen die Autoren die Frage, ob diese Algorithmen so auf den Gitterfall übertragen werden können, dass eine polynomiale Gitterschranke resultiert. Hinsichtlich der Entwicklung von Algorithmen für das Erstellen von Gitterzeichnungen mit niedriger Komplexität ist demnach aktuelles Forschungsinteresse vorhanden. Auch besteht ein eigenständiges Erkenntnisinteresse an solchen Algorithmen. Darüber hinaus würden sie die Möglichkeit bieten, den Zusammenhang zwischen visueller Komplexität und Lesbarkeit eines Graphen weiter zu erforschen. Wie zuvor beschrieben sind erst zwei Algorithmen dieser Art bekannt. In der vorliegenden Arbeit sollen daher Algorithmen entwickelt werden, die für verschiedene Graphenklassen Gitterzeichnungen auf einem polynomialen Gitter mit niedriger visueller Komplexität erstellen. Die Kanten sollen entsprechend dem meistverbreiteten Zeichenstil mit geradlinigen Segmenten gezeichnet werden. Auch sollen die Kanten kreuzungsfrei sein. Es kommen daher nur planare Graphen in Betracht. Konkret sollen Bäume, 3-Bäume, serien-parallele Graphen und außenplanare Graphen untersucht werden. Für Bäume bietet sich dabei eine Modifizierung des Algorithmus aus [41] an. Ferner sollen die entworfenen Algorithmen auch prototypisch implementiert werden.

## 2. Gitterzeichnungen mit niedriger visueller Komplexität

Zunächst werden zwei Bezeichnungen eingeführt. Eine Zeichnung eines Graphen, die mit der für die entsprechende Graphenklasse geltenden minimalen Anzahl von Segmenten angefertigt worden ist, wird als segmentminimale Zeichnung bezeichnet. Eine Zeichnung eines Graphen, die mit einer im Vergleich zur Kantenzahl verminderten Segmentanzahl, jedoch nicht mit der minimalen Segmentanzahl gezeichnet worden ist, wird als segmentreduzierte Zeichnung bezeichnet. Die auf die konkrete Themenstellung hin entworfenen Algorithmen werden in den folgenden Unterkapiteln beschrieben. Nach Beweis ihrer Korrektheit wird jeweils die visuelle Komplexität und Gitterschranke der von ihnen erstellten Gitterzeichnungen analysiert. Die Beschreibung beginnt mit Bäumen, die eine der wichtigsten Graphenklassen bilden und strukturell einfach sind. Es folgen 3-Bäume, bevor außenplanare und serien-parallele Graphen den Abschluss bilden.

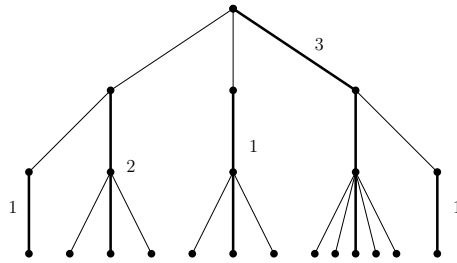
### 2.1. Segmentreduzierte Gitterzeichnungen von Bäumen

Bäume sind genau die zusammenhängenden kreisfreien Graphen. Ein Test auf Zusammenhang und Kreisfreiheit ist ein einfacher Erkennungsalgorithmus für Bäume. Da viele Sachverhalte hierarchisch strukturiert sind, haben Bäume erhebliche Bedeutung in der Visualisierung von Informationen. Zunächst wird im Grundlagenabschnitt mit der Zerlegung eines Baums in schwere Pfade eine Dekompositionstechnik beschrieben, die wesentliche Grundlage für den im weiteren beschriebenen Algorithmus ist. Dieser Algorithmus erstellt segmentreduzierte Zeichnungen von Bäumen und verwendet wie [41] das Konzept der sicheren Box. Da anstatt Kreisbögen wie in [41] jetzt gerade Liniensegmente verwendet werden, erhalten die sicheren Boxen ein anderes Design. Entsprechend muss beim rekursiven Aufbau der sicheren Boxen eine andere Strategie verfolgt werden.

#### 2.1.1. Spezifische Grundlagen

**Definition 1 ([42][41]):** Die Zerlegung eines Baums  $T$  in schwere Pfade ist wie folgt definiert: Man bestimme einen (beliebigen) Knoten  $r$  von  $T$  zur Wurzel. In einem vorbereitenden Durchlauf durch  $T$  wird für jeden Knoten  $v$  die Anzahl der Knoten in dem an  $v$  verwurzelten Unterbaum  $T_v$  berechnet. Sei  $N(v)$  die Anzahl dieser Knoten beziehungsweise die Größe von  $T_v$ . Man bestimme nun für jeden Knoten  $v$ , der kein Blatt ist, denjenigen Kindknoten  $u$ , für den—bezogen auf die Größe der Unterbäume aller Kindknoten— $N(u)$  maximal ist. Dann nennt man die Kante  $(v, u)$  eine schwere Kante. Die von  $v$  zu den übrigen Kindknoten verlaufenden Kanten sind sogenannte leichte Kanten. Eine maximale verbundene Komponente von schweren Kanten wird ein schwerer Pfad genannt. Hierdurch wird ein Baum in schwere Pfade und leichte Kanten zerlegt (Abbildung 1). Es wird folgende Terminologie verwendet:

- (1) Derjenige Knoten eines schweren Pfads, der am nächsten zur Wurzel gelegen ist, wird Topknoten genannt.
- (2) Der am Topknoten eines schweren Pfads verwurzelte Unterbaum wird als Unterbaum dieses schweren Pfads bezeichnet.
- (3) Die Kante, die den Topknoten eines schweren Pfads  $P$  mit seinem Elternknoten in  $T$  verbindet, wird leichte Elternkante von  $P$  genannt.
- (4) Die Tiefe eines schweren Pfads  $P$  wird rekursiv definiert.
  - (a) Wenn  $P$  nicht inzident zu leichten Elternkanten von anderen schweren Pfaden ist, hat  $P$  die Tiefe eins.
  - (b) Andernfalls wird die größte Tiefe der schweren Pfade ermittelt, die durch eine leichte Elternkante mit  $P$  verbunden sind. Der so ermittelte Wert wird um eins inkrementiert und  $P$  als Tiefe zugewiesen.



**Abbildung 1:** Zerlegung eines Baums in schwere Pfade und leichte Kanten einschließlich annotierter Tiefe der schweren Pfade (in Anlehnung an [41]).

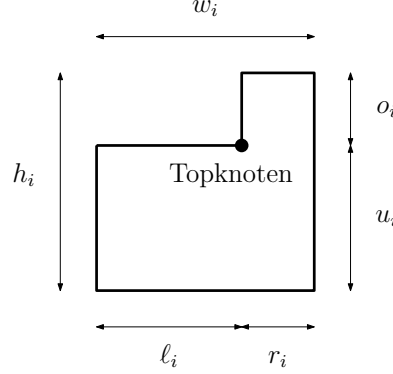
**Hilfssatz 1 ([42][41]):** Sei  $n$  die Anzahl der Knoten eines Baums  $T$ , für den eine Zerlegung in schwere Pfade durchgeführt wurde. Dann hat jeder Weg zur Wurzel des Baums höchstens  $\lceil \log_2 n \rceil$  leichte Elternkanten.

**Hilfssatz 2 ([41]):** Die Unterbäume von schweren Pfaden gleicher Tiefe sind disjunkt.

### 2.1.2. Algorithmus TREEFEWSEG

**Beschreibung:** Mittels Algorithmus TREEFEWSEG wird ein gegebener Baum  $T$  mit einer verminderten Anzahl von Segmenten gezeichnet. Hierbei ist der übergeordnete Programmablauf analog zu [41]. Zunächst wird eine Zerlegung von  $T$  in schwere Pfade berechnet. Die Unterbäume dieser schweren Pfade werden dann—bezogen auf die Tiefe ihres zugrunde liegenden schweren Pfads—in aufsteigender Ordnung gezeichnet. Der Algorithmus endet dann mit der vollständigen Zeichnung von  $T$ , wenn der Unterbaum von dem schweren Pfad mit der größten Tiefe gezeichnet ist. Im Algorithmus wird das Konzept der sicheren Box verwendet. Dabei wird einem Unterbaum eine Box zugeordnet. Die in der Box gezeichneten Knoten des Unterbaums ändern im weiteren Verlauf des Algorithmus ihre Position relativ zur sicheren Box nicht mehr. Grundsätzlich wird die sichere Box eines Unterbaums rekursiv aus den Boxen von Unterbäumen solcher schweren Pfade zusammengesetzt, die eine geringere Tiefe haben als der schwere Pfad des Unterbaums, dessen Box gerade konstruiert wird.

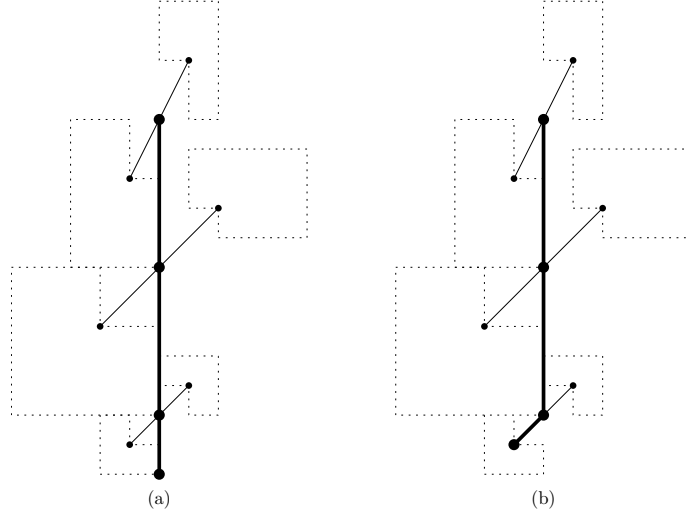
Aufbau und Abmessungen einer sicheren Box  $B_i$  lassen sich der Abbildung 2 entnehmen. Als einziger Knoten befindet sich der Topknoten auf der Kontur der Box, und zwar an der spitzen Ecke der Box. Die Weite der Box beträgt  $w_i = l_i + r_i$  und die Höhe  $h_i = o_i + u_i$ .



**Abbildung 2:** Eine sichere Box und ihre Abmessungen.

Ein einzelnes Blatt des Baums, das zu keinem schweren Pfad inzident ist, wird in einer sicheren Box der Höhe zwei und der Breite zwei positioniert. Formal betrachtet man im Hinblick auf die Zeichenfolge der Boxen ein solches Blatt als einen schweren Pfad der Tiefe null. Zusätzlich gelte für alle Boxen die Invariante  $u_i \geq o_i$ . Die Boxen können im Verlauf des Algorithmus horizontal oder vertikal gespiegelt werden. Man nehme an, dass alle Boxen der Unterbäume von schweren Pfaden der Tiefe  $\ell$  konstruiert sind und alle Boxen den beschriebenen Aufbau haben. Es sollen nun die Boxen der Unterbäume von schweren Pfaden der Tiefe  $(\ell + 1)$  konstruiert werden.

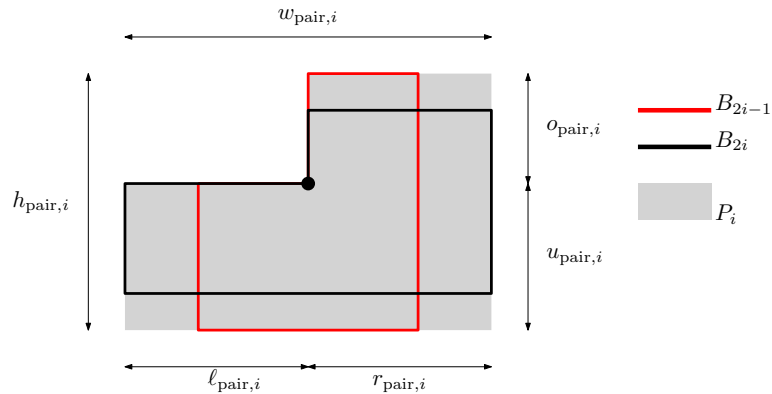
Die Konstruktion einer solchen Box  $B_{\text{neu}}$  verläuft in Runden. Dabei wird pro Runde ein Knoten des zugehörigen schweren Pfads verarbeitet. In einer solchen Runde werden die Boxen von denjenigen Unterbäumen positioniert, die über eine leichte Elternkante mit diesem Knoten verbunden sind. Dies geschieht, indem jeweils zwei dieser leichten Elternkanten mittels eines einzigen Segments gezeichnet werden. Für den schweren Pfad einer solchen Box werden ein oder zwei Segmente benötigt. Dies ist von der Anzahl der Kindknoten des vorletzten Knotens des schweren Pfads abhängig. Alle diese Kindknoten sind Blätter in  $T$ , da andernfalls der gemeinsame Elternknoten nicht vorletzter Knoten eines schweren Pfads sein könnte. Ist die Anzahl dieser Kindknoten ungerade, wird der schwere Pfad mit einem Segment gezeichnet (Abbildung 3(a)). Dabei wird der letzte Knoten des schweren Pfads auf der Kontur der Box eines der anderen Blätter positioniert. Ist die Anzahl der betrachteten Kindknoten gerade, werden zwei Segmente für den schweren Pfad verwendet. In diesem Fall wird der letzte Knoten des schweren Pfads als Topknoten in einer Box der Weite zwei und Höhe zwei positioniert (Abbildung 3(b)). In [41] wird die Anordnung der Kindknoten des vorletzten Knotens des schweren Pfads mit dem Begriff *k-fork* beschrieben. Die Anzahl der Segmente für die Zeichnung des schweren Pfads hängt dann davon ab, ob  $k$  gerade oder ungerade ist. Der letzte Knoten des schweren Pfads verursacht keine eigene Runde. Er wird—wie zuvor beschrieben—bereits in der Runde des vorletzten Knotens verarbeitet.



**Abbildung 3:** (a) Zeichnung des schweren Pfads, wenn der vorletzte Knoten des schweren Pfads eine ungerade Anzahl von Knoten hat. (b) Zeichnung des schweren Pfads, wenn der vorletzte Knoten des schweren Pfads eine gerade Anzahl von Knoten hat.

In der ersten Runde wird der Topknoten des zugehörigen schweren Pfads behandelt. Sei  $q_0$  dieser Topknoten. Sei ab jetzt mit  $k$  die Anzahl der Unterbäume bezeichnet, die über ihre leichte Elternkante mit  $q_0$  verbunden sind. Dann ist die Anzahl der in der ersten Runde zu positionierenden Boxen ebenfalls gleich  $k$ .

Wenn  $k$  gerade ist, werden die Boxen so sortiert, dass für  $1 \leq i \leq k/2$  die Relation  $u_{2i-1} \geq u_{2i}$  gilt. Aus den Boxen  $B_{2i-1}$  und  $B_{2i}$  wird nun für  $1 \leq i \leq k/2$  die Paarbox  $P_i$  gebildet. Dabei sind die Topknoten von  $B_{2i-1}$  und  $B_{2i}$  koinzident. Einer Paarbox  $P_i$  werden die Abmessungen  $l_{\text{pair},i} = \max(l_{2i}, l_{2i-1})$ ,  $r_{\text{pair},i} = \max(r_{2i}, r_{2i-1})$ ,  $o_{\text{pair},i} = \max(o_{2i}, o_{2i-1})$ ,  $u_{\text{pair},i} = \max(u_{2i}, u_{2i-1})$  und  $w_{\text{pair},i} = l_{\text{pair},i} + r_{\text{pair},i}$ ,  $h_{\text{pair},i} = o_{\text{pair},i} + u_{\text{pair},i}$  zugewiesen (Abbildung 4).

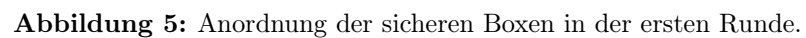


**Abbildung 4:** Eine Paarbox und ihre Abmessungen.

Wenn  $k$  ungerade ist, bildet man formal die letzte Paarbox  $P_{\lceil k/2 \rceil}$  aus der Box  $B_k$  und einer zusätzlichen virtuellen Box  $B_{k+1}$ , die keine Knoten—auch keinen Topknoten—beinhaltet und keine Ausdehnung hat. Es gilt also  $h_{k+1} = w_{k+1} = o_{k+1} = u_{k+1} = l_{k+1} = r_{k+1} = 0$ . Die Abmessungen von  $P_{\lceil k/2 \rceil}$  und  $B_k$  stimmen somit überein.



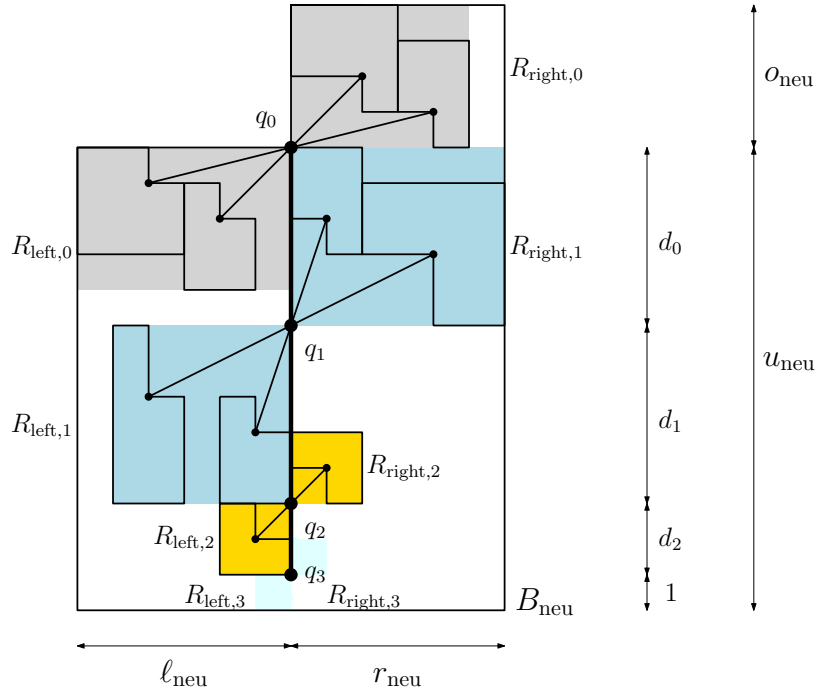
Die Paarbox  $P_1$  und ihre Kopie liegen am äußeren Ende dieses Arrangements. Die letzte Paarbox  $P_{\lceil k/2 \rceil}$  und ihre Kopie sind benachbart zum schweren Pfad bzw. seiner Verlängerung über  $q_0$  hinaus. Die obere (untere) Kante von  $P_1$  (von der Kopie von  $P_1$ ) liegt auf gleichen Zeile wie  $q_0$ . Bei allen weiteren Paarboxen liegt die obere (untere) Kante von  $P_i$  (von der Kopie von  $P_i$ ) auf der gleichen Zeile wie der Topknoten von  $P_{i-1}$  (der Kopie von  $P_{i-1}$ ). Für  $1 \leq i \leq k/2$  wird nun die links unterhalb von  $q_0$  gelegenen Paarbox  $P_i$  durch die Box  $B_{2i-1}$  und die rechts oberhalb von  $q_0$  gelegene Kopie von  $P_i$  durch  $B_{2i}$  ersetzt. Dies geschieht so, dass die spitze Ecke von  $B_{2i-1}$  ( $B_{2i}$ ) mit der spitzen Ecke von  $P_i$  (der Kopie von  $P_i$ ) übereinstimmt. Wegen  $u_{2i-1} \geq u_{2i}$  ist damit gewährleistet, dass für die neu zu konstruierende Box  $B_{\text{neu}}$  die Relation  $u_{\text{neu}} \geq o_{\text{neu}}$  gilt und somit die Invariante  $u \geq o$  auch für  $B_{\text{neu}}$  gültig ist. Letztendlich sind der Topknoten von  $B_{2i-1}$ , der Knoten  $q_0$  und der Topknoten von  $B_{2i}$  kollinear. Daher können die leichten Elternkanten von  $B_{2i-1}$  und  $B_{2i}$  mit einem einzigen Segment gezeichnet werden. Nachdem die sicheren Boxen mit Hilfe der Paarboxen so positioniert wurden, werden die Paarboxen nicht mehr benötigt und bei der Konstruktion nicht weiter betrachtet.



11

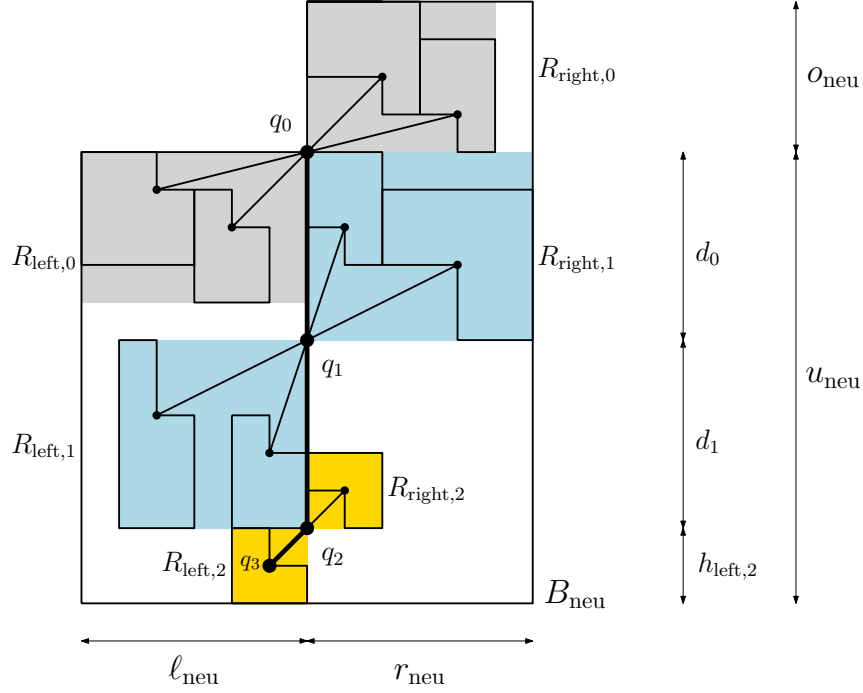
wie die Abstände der Knoten auf dem schweren Pfad bestimmt werden. Sei  $p \geq 1$  die Anzahl der Knoten des schweren Pfads. Mit  $q_s, 0 \leq s \leq p$  seien die Knoten des schweren Pfads bezeichnet. Mit  $R_{\text{left},s}$  sei dann das Rechteck bezeichnet, das das Arrangement derjenigen sicheren Boxen umhüllt, die über ihre leichte Elternkante mit  $q_s$  verbunden sind und die links unterhalb von  $q_s$  liegen. Entsprechend sei mit  $R_{\text{right},s}$  das Rechteck bezeichnet, das das Arrangement derjenigen sicheren Boxen umhüllt, die über ihre leichte Elternkante mit  $q_s$  verbunden sind und die rechts oberhalb von  $q_s$  liegen. Mit  $h_{\text{left},s}$ ,  $h_{\text{right},s}$ ,  $w_{\text{left},s}$  und  $w_{\text{right},s}$  werden Höhe und Weite der beiden zu  $q_s$  inzidenten Rechtecke bezeichnet. Nachdem zu Beginn der ersten Runde  $q_0$  gezeichnet worden ist, wird zu Beginn einer weiteren Runde der jeweils nächste Knoten des schweren Pfads so positioniert, dass zwischen  $q_s$  und  $q_{s+1}$  ein vertikaler Abstand von  $d_s = \max(h_{\text{left},s}, h_{\text{right},s+1})$  vorhanden ist. Die Kontur von  $B_{\text{neu}}$  wird nun so gewählt, dass  $q_0$  als Topknoten an der spitzen Ecke der Box liegt und keine weiteren Knoten auf der Kontur liegen. Die Gültigkeit der Invariante  $u \geq o$  für  $B_{\text{neu}}$  wurde bereits zuvor nachgewiesen. Hilfssatz 2 garantiert, dass die Boxen der Unterbäume von schweren Pfaden gleicher Tiefe disjunkt sind und somit widerspruchsfrei konstruiert werden können. Im Detail wird hinsichtlich der Abmessungen von  $B_{\text{neu}}$  entsprechend der Fallunterscheidung für das Zeichnen des schweren Pfads wie folgt unterschieden:

- (1) Der vorletzte Knoten des schweren Pfads hat eine ungerade Anzahl von Kindknoten. Dem letzten Knoten  $q_p$  des schweren Pfads werden in diesem Fall formal zwei Rechtecke  $R_{\text{left},p}$  und  $R_{\text{right},p}$  zugeordnet, für die  $h_{\text{left},p} = 1$ ,  $h_{\text{right},p} = 1$ ,  $w_{\text{left},p} = 1$ ,  $w_{\text{right},p} = 1$  gilt und die keine sicheren Boxen beinhalten. Dann beträgt der untere Abschnitt der neuen Box  $u_{\text{neu}} = \sum_{s=0}^{p-2} d_s + d_{p-1} + 1 = \sum_{s=0}^{p-2} d_s + h_{\text{left},p-1} + 1$  (Abbildung 6).



**Abbildung 6:** Eine neu konstruierte Box  $B_{\text{neu}}$  und ihre Abmessungen, wenn der vorletzte Knoten des schweren Pfads eine ungerade Anzahl von Kindknoten hat.

- (2) Der vorletzte Knoten des schweren Pfads hat eine gerade Anzahl von Kindknoten. Dann beträgt der untere Abschnitt der neuen Box  $u_{\text{neu}} = \sum_{s=0}^{p-2} d_s + h_{\text{left},p-1}$  (Abbildung 7).



**Abbildung 7:** Eine neu konstruierte Box  $B_{\text{neu}}$  und ihre Abmessungen, wenn der vorletzte Knoten des schweren Pfads eine gerade Anzahl von Kindknoten hat.

In beiden Fällen gilt für die weiteren Abmessungen  $l_{\text{neu}} = \max_s(w_{\text{left},s})$ ,  $r_{\text{neu}} = \max_s(w_{\text{right},s})$  und  $o_{\text{neu}} = h_{\text{right},0}$ . Eine konkrete Implementierung muss den Sonderfall berücksichtigen, dass überhaupt keine sichere Box rechts oberhalb von  $q_0$  positioniert wird.

Damit ist der Konstruktionsprozess von  $B_{\text{neu}}$  vollständig beschrieben. Der Algorithmus selber terminiert nach Zeichnung der sicheren Box des Unterbaums, dessen schwerer Pfad die größte Tiefe hat. Diese Zeichnung ist eine vollständige segmentreduzierte Gitterzeichnung von  $T$ . Ihre Gittergröße wird nun analysiert.

**Gittergröße:** Es werden zunächst die Höhe  $h_0$  und die Weite  $w_0$  berechnet, die zusammen all die sicheren Boxen einnehmen, die über ihre leichte Elternkante mit  $q_0$  verbunden sind. Dabei setzen sich  $h_0$  und  $w_0$  jeweils aus zwei Anteilen zusammen, es gilt  $h_0 = h_{\text{left},0} + h_{\text{right},0}$  und  $w_0 = w_{\text{left},0} + w_{\text{right},0}$ . Die Anzahl der zu betrachtenden sicheren Boxen wurde bereits zuvor mit  $k$  bezeichnet.

Für die weitere Analyse nehme man zunächst an, dass  $k$  gerade ist. In diesem Fall berechnet man  $h_{\text{left},0}$ , indem für jede Box  $B_{2j-1}$ ,  $1 \leq j \leq k/2$  der Wert  $u_{2j-1} + \sum_{i=1}^j o_{\text{pair},i}$  bestimmt und das Maximum aus diesen Werten gebildet wird. Analog berechnet man im Hinblick auf  $h_{\text{right},0}$  für jede Box  $B_{2j}$ ,  $1 \leq j \leq k/2$  den Wert  $u_{2j} + \sum_{i=1}^j o_{\text{pair},i}$  und daraus wieder das Maximum dieser Werte. Dann erhält man mit den zuvor getätigten

Bezeichnungen und der Abbildung 5 die folgende Abschätzung für  $h_0$  und  $w_0$ —weiterhin unter der Annahme, dass  $k$  gerade ist:

$$\begin{aligned}
h_0 &= h_{\text{left},0} + h_{\text{right},0} \\
h_0 &= \max_{1 \leq j \leq k/2} (u_{2j-1} + \sum_{i=1}^j o_{\text{pair},i}) + \max_{1 \leq j \leq k/2} (u_{2j} + \sum_{i=1}^j o_{\text{pair},i}) \\
h_0 &= \max_{1 \leq j \leq k/2} (u_{2j-1} + \sum_{i=1}^j \max(o_{2i-1}, o_{2i})) + \max_{1 \leq j \leq k/2} (u_{2j} + \sum_{i=1}^j \max(o_{2i-1}, o_{2i}))
\end{aligned}$$

Wegen  $o_i \geq 0$  kann man die Abschätzung  $\max(o_{2i-1}, 2i) \leq o_{2i-1} + o_{2i}$  anwenden und erhält:

$$\begin{aligned}
h_0 &\leq \max_{1 \leq j \leq k/2} (u_{2j-1} + \sum_{i=1}^j (o_{2i-1} + o_{2i})) + \max_{1 \leq j \leq k/2} (u_{2j} + \sum_{i=1}^j (o_{2i-1} + o_{2i})) \\
h_0 &\leq \max_{1 \leq j \leq k/2} (u_{2j-1} + \sum_{i=1}^{k/2} (o_{2i-1} + o_{2i})) + \max_{1 \leq j \leq k/2} (u_{2j} + \sum_{i=1}^{k/2} (o_{2i-1} + o_{2i})) \\
h_0 &\leq \max_{1 \leq j \leq k/2} (u_{2j-1} + \sum_{i=1}^k o_i) + \max_{1 \leq j \leq k/2} (u_{2j} + \sum_{i=1}^k o_i) \\
h_0 &\leq \max_{1 \leq j \leq k/2} u_{2j-1} + \max_{1 \leq j \leq k/2} u_{2j} + \sum_{i=1}^k o_i + \sum_{i=1}^k o_i \\
h_0 &\leq \sum_{i=1}^k u_i + 2 \sum_{i=1}^k o_i.
\end{aligned}$$

Da für alle  $i$  mit  $1 \leq i \leq k$  die Relation  $u_i \geq o_i$  gilt, folgt  $\sum_{i=1}^k o_i \leq \sum_{i=1}^k u_i$  und somit:

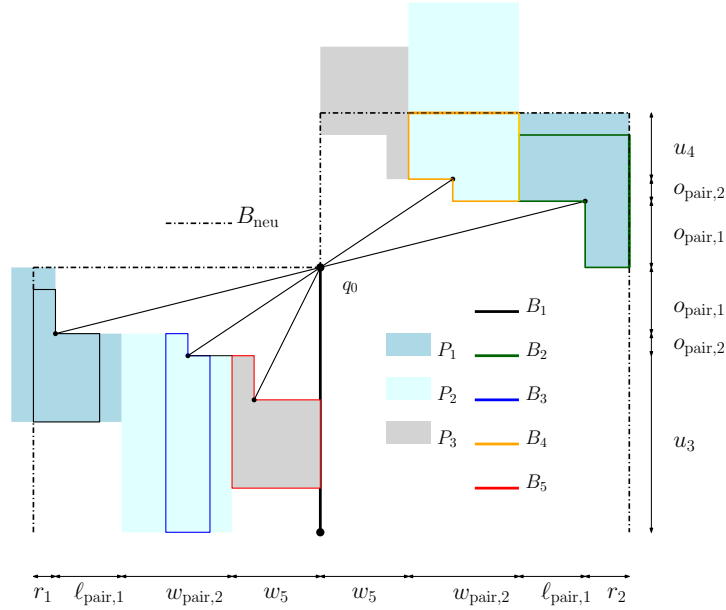
$$\begin{aligned}
h_0 &\leq \frac{3}{2} \sum_{i=1}^k u_i + \frac{3}{2} \sum_{i=1}^k o_i \text{ bzw. mit } h_i = o_i + u_i \\
h_0 &\leq \frac{3}{2} \sum_{i=1}^k h_i.
\end{aligned}$$

Für  $w_0$  erhält man mit Abbildung 5:

$$\begin{aligned}
w_0 &= w_{\text{left},0} + w_{\text{right},0} \\
w_0 &= (r_1 + l_{\text{pair},1} + \sum_{i=2}^{k/2} w_{\text{pair},i}) + (\sum_{i=2}^{k/2} w_{\text{pair},i} + l_{\text{pair},1} + r_2) \\
w_0 &\leq r_{\text{pair},1} + l_{\text{pair},1} + 2 \sum_{i=2}^{k/2} (l_{\text{pair},i} + r_{\text{pair},i}) + l_{\text{pair},1} + r_{\text{pair},1} \\
w_0 &\leq 2 \sum_{i=1}^{k/2} (l_{\text{pair},i} + r_{\text{pair},i}) \\
w_0 &\leq 2 \sum_{i=1}^{k/2} \max(l_{2i}, l_{2i-1}) + 2 \sum_{i=1}^{k/2} \max(r_{2i}, r_{2i-1})
\end{aligned}$$

$$\begin{aligned}
w_0 &\leq 2 \sum_{i=1}^{k/2} (l_{2i} + l_{2i-1}) + 2 \sum_{i=1}^{k/2} (r_{2i} + r_{2i-1}) \\
w_0 &\leq 2 \sum_{i=1}^k l_i + 2 \sum_{i=1}^k r_i \\
w_0 &\leq 2 \sum_{i=1}^k w_i \text{ mit } w_i = l_i + r_i.
\end{aligned}$$

Sei  $k$  nun ungerade. Ein Beispiel für  $k = 5$  zeigt Abbildung 8:



**Abbildung 8:** Abmessungen von  $B_{neu}$  bei ungerader Anzahl der Boxen.

Wie bei der Konstruktion der Paarbox auf Seite 10 beschrieben, existiert die Paarbox  $P_{\lceil k/2 \rceil}$  und wird von  $B_k$  und der virtuellen Box  $B_{k+1}$  gebildet. Da  $B_{k+1}$  keine Ausdehnung hat, stimmt  $P_{\lceil k/2 \rceil}$  in ihren Abmessungen mit  $B_k$  überein. Die Abschätzung von  $w_0$  kann ohne die Verwendung von  $P_{\lceil k/2 \rceil}$  erfolgen und lautet:

$$\begin{aligned}
w_0 &= w_{\text{left},0} + w_{\text{right},0} \\
w_0 &= r_1 + l_{\text{pair},1} + \sum_{i=2}^{\lfloor k/2 \rfloor} w_{\text{pair},i} + w_k + w_k + \sum_{i=2}^{\lfloor k/2 \rfloor} w_{\text{pair},i} + l_{\text{pair},1} + r_2 \\
w_0 &\leq r_{\text{pair},1} + l_{\text{pair},1} + 2 \sum_{i=2}^{\lfloor k/2 \rfloor} (l_{\text{pair},i} + r_{\text{pair},i}) + 2w_k + l_{\text{pair},1} + r_{\text{pair},1} \\
w_0 &\leq 2 \sum_{i=1}^{\lfloor k/2 \rfloor} (l_{\text{pair},i} + r_{\text{pair},i}) + 2w_k \\
w_0 &\leq 2 \sum_{i=1}^{\lfloor k/2 \rfloor} \max(l_{2i}, l_{2i-1}) + 2 \sum_{i=1}^{\lfloor k/2 \rfloor} \max(r_{2i}, r_{2i-1}) + 2w_k
\end{aligned}$$

$$\begin{aligned}
w_0 &\leq 2 \sum_{i=1}^{\lfloor k/2 \rfloor} (l_{2i} + l_{2i-1}) + 2 \sum_{i=1}^{\lfloor k/2 \rfloor} (r_{2i} + r_{2i-1}) + 2(l_k + r_k) \\
w_0 &\leq 2(l_k + \sum_{i=1}^{k-1} l_i) + 2(r_k + \sum_{i=1}^{k-1} r_i) \\
w_0 &\leq 2 \sum_{i=1}^k l_i + 2 \sum_{i=1}^k r_i \\
w_0 &\leq 2 \sum_{i=1}^k w_i \text{ mit } w_i = l_i + r_i.
\end{aligned}$$

In der Abschätzung von  $h_0$  werden zunächst für jede sichere Box  $B_j$  die oberen Abschnitte von Paarboxen addiert—beginnend mit der ersten Paarbox und endend mit der Paarbox, der  $B_j$  angehört. Um diese Addition für alle sicheren Boxen einschließlich  $B_k$  einheitlich darstellen zu können, wird die Paarbox  $P_{\lfloor k/2 \rfloor}$  verwendet. Es gilt:

$$\begin{aligned}
h_0 &= h_{\text{left},0} + h_{\text{right},0} \\
h_0 &= \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j-1} + \sum_{i=1}^j o_{\text{pair},i}) + \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j} + \sum_{i=1}^j o_{\text{pair},i}) \\
h_0 &= \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j-1} + \sum_{i=1}^j \max(o_{2i-1}, o_{2i})) + \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j} + \sum_{i=1}^j \max(o_{2i-1}, o_{2i})) \\
h_0 &\leq \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j-1} + \sum_{i=1}^j (o_{2i-1} + o_{2i})) + \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j} + \sum_{i=1}^j (o_{2i-1} + o_{2i})) \\
h_0 &\leq \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j-1} + \sum_{i=1}^{\lfloor k/2 \rfloor} (o_{2i-1} + o_{2i})) + \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j} + \sum_{i=1}^{\lfloor k/2 \rfloor} (o_{2i-1} + o_{2i})) \\
h_0 &\leq \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j-1} + (o_k + o_{k+1}) + \sum_{i=1}^{\lfloor k/2 \rfloor} (o_{2i-1} + o_{2i})) + \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j} - o_k + \sum_{i=1}^k o_i)
\end{aligned}$$

Mit  $o_{k+1} = 0$  und  $o_k > 0$  folgt:

$$\begin{aligned}
h_0 &\leq \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j-1} + \sum_{i=1}^k o_i) + \max_{1 \leq j \leq \lfloor k/2 \rfloor} (u_{2j} + \sum_{i=1}^k o_i) \\
h_0 &\leq \max_{1 \leq j \leq \lfloor k/2 \rfloor} u_{2j-1} + \max_{1 \leq j \leq \lfloor k/2 \rfloor} u_{2j} + \sum_{i=1}^k o_i + \sum_{i=1}^k o_i \\
h_0 &\leq \sum_{i=1}^k u_i + 2 \sum_{i=1}^k o_i
\end{aligned}$$

Man verwendet wieder die Relation  $\sum_{i=1}^k o_i \leq \sum_{i=1}^k u_i$  und erhält wie im ersten Fall:

$$h_0 \leq \frac{3}{2} \sum_{i=1}^k h_i.$$

Die für eine gerade bzw. ungerade Anzahl von sicheren Boxen erzielten Abschätzungen stimmen somit sowohl für  $h_0$  als auch für  $w_0$  überein.

Da in den weiteren Hierarchieebenen des Unterbaums die Positionierung der Boxen auf die gleiche Art wie für  $q_0$  erfolgt, gelten für die weiteren Level des Unterbaums dieselben Abschätzungen. Ein schwerer Pfad der Tiefe null wird in einer sicheren Box der Weite zwei und der Höhe zwei gezeichnet. Aufgrund der gewonnenen Abschätzungen befindet sich daher ein schwerer Pfad der Tiefe  $d$  mit  $m$  Knoten in einer Box, deren Weite höchstens  $w = 2 \cdot 2^d \cdot m$  und deren Höhe höchstens  $h = 2 \cdot (3/2)^d \cdot m$  beträgt. Da nach Hilfssatz 1 die Rekursionstiefe für einen Baum mit  $n$  Knoten höchstens  $\lceil \log_2 n \rceil$  beträgt, wird ein Baum mit  $n$  Knoten in einer Box gezeichnet, deren Weite höchstens  $w = 2 \cdot 2^{\lceil \log_2 n \rceil} \cdot n$  und deren Höhe höchstens  $h = 2 \cdot (3/2)^{\lceil \log_2 n \rceil} \cdot n$  ist. Als asymptotische Gitterschranke ergibt sich daher für die Weite  $O(n^{1+\log_2 3/2}) = O(n^{1,58})$  und für die Höhe  $O(n^{1+\log_2 2}) = O(n^2)$ .

**Visuelle Komplexität:** Es besteht Übereinstimmung zwischen dem Algorithmus aus [41] und Algorithmus TREEFEWSEG hinsichtlich der Zeichnung des schweren Pfads, dem rekursiven Aufbau der Boxen, der Ausführung in Runden und der Zeichnung von je zwei leichten Elternkanten mit einem einzigen Segment. Die Analyse der visuellen Komplexität kann somit ohne Änderungen aus [41] übernommen werden. Ein Baum  $T = (V, E)$  wird daher durch Algorithmus TREEFEWSEG mit höchstens  $\lceil 3|E|/4 \rceil$  Segmenten gezeichnet.

### 2.1.3. Zusammenfassung

Die gewonnenen Ergebnisse lassen sich im folgenden Satz 1 zusammenfassen:

**Satz 1:** Ein Baum  $T = (V, E)$  mit  $|V| = n$  wird durch Algorithmus TREEFEWSEG auf einem Gitter der Größe  $O(n^2) \times O(n^{1,58})$  in  $\lceil 3|E|/4 \rceil$  Segmenten gezeichnet.

## 2.2. Segmentreduzierte Gitterzeichnungen von 3-Bäumen

Der in diesem Kapitel beschriebene Algorithmus für die Anfertigung von segmentreduzierten Gitterzeichnungen von 3-Bäumen verwendet die Eigenschaft von 3-Bäumen, auch ebene Dreiecksgraphen zu sein. Sie haben daher eine kanonische Ordnung und einen Schnyder-Realisierer. Die Eigenschaften dieser beiden Zerlegungen, die Definition von allgemeinen  $k$ -Bäumen und die induktive Deutung eines 3-Baums bilden im Wesentlichen den Grundlagenteil dieses Kapitels. Danach wird ein Algorithmus beschrieben, der von einem 3-Baum eine segmentreduzierte Zeichnung auf einem polynomialen Gitter erstellt. In dem sich anschließenden umfangreichen Induktionsbeweis wird gezeigt, dass der Algorithmus korrekt ist.

### 2.2.1. Spezifische Grundlagen

**Definition 2 ([37]):** Ein Graph  $G$  ist ein  $k$ -Baum genau dann, wenn

- (1)  $G$  zusammenhängend ist,
- (2)  $G$  eine  $k$ -Clique, aber keine  $(k+2)$ -Clique hat,
- (3) jeder minimale  $X, Y$ -Separator von  $G$  eine  $k$ -Clique ist.

**Definition 3 ([35]):** Eine rekursive Definition von  $k$ -Bäumen lautet:

- (1) Der vollständige Graph  $K_k$  ist ein  $k$ -Baum.
- (2) Einen  $k$ -Baum mit  $(n+1)$  Knoten erhält man aus einem  $k$ -Baum  $T$  mit  $n$  Knoten durch das Hinzufügen eines neuen Knotens, der genau zu einer  $k$ -Clique in  $T$  benachbart ist.

**Definition 4:** Speziell ein 3-Baum kann auch als Stapelgraph gedeutet werden. Definition 3 ist dann für  $k = 3$  gleichbedeutend mit der bekannten rekursiven Definition eines 3-Baums.

- (1) Ein ebenes Dreieck stellt den 3-Baum mit drei Knoten dar.
- (2) Einen 3-Baum mit  $(n+1)$  Knoten erhält man aus einem 3-Baum  $T$  mit  $n$  Knoten, indem man einen neuen Knoten auf einer inneren Dreiecksfläche von  $T$  positioniert und mit den drei Eckpunkten des Dreiecks verbindet.

Diese sogenannte Stapeloperation ahmt die induktive Definition eines 3-Baums nach.

**Definition 5 ([20] nach der Darstellung aus [14]):** Eine kanonische Ordnung eines ebenen Dreiecksgraphen ist wie folgt definiert:

Sei  $G$  ein ebener Dreiecksgraph mit  $n$  Punkten und  $(v_1, v_2, v_n)$  die Außenfläche von  $G$  gegen den Uhrzeigersinn. Sei  $\sigma = (v_1, v_2, \dots, v_n)$  eine Ordnung aller Knoten von  $G$  und  $G_k$ ,  $3 \leq k \leq n$  der durch  $v_1 \cup v_2 \cup \dots \cup v_k$  induzierte Untergraph. Mit  $P_k$  sei der im Uhrzeigersinn verlaufende Weg auf der Außenfläche von  $G$  bezeichnet, der in  $v_1$  startet und in  $v_2$  endet. Man nennt  $\sigma$  eine kanonische Ordnung von  $G$  mit Bezug auf die äußere Kante  $(v_1, v_2)$ , wenn für jedes  $k$ ,  $3 \leq k \leq n$  die folgenden Bedingungen erfüllt sind:

- (1) Der Graph  $G_k$  ist 2-zusammenhängend und im Inneren trianguliert.
- (2) Wenn  $k+1 \leq n$ , dann ist  $v_{k+1}$  ein äußerer Knoten von  $G_{k+1}$  und die Nachbarn von  $v_{k+1}$  in  $G_k$  erscheinen fortlaufend auf  $P_k$ .

**Definition 6:** Der in Definition 5 definierte Weg  $P_k$  auf der Außenfläche von  $G_k$  wird im Folgenden auch als Kontur bezeichnet, geschrieben als  $C_k = (w_1 = v_1, w_2, \dots, w_m = v_2)$ .

**Definition 7:** Es seien  $w_p, \dots, w_q$  die gemäß Definition 5 fortlaufend auf dem Weg  $P_k$  erscheinenden Nachbarn von  $v_{k+1}$  in  $G_k$ .

- (1) Der Knoten  $w_p$  wird der ganz linke Vorfahre,  $w_q$  der ganz rechte Vorfahre von  $v_{k+1}$  genannt.
- (2) Die Kante  $(v_{k+1}, w_p)$  wird als ganz linke Kante von  $V_k$  bezeichnet, die Kante  $(v_{k+1}, w_q)$  als ganz rechte Kante von  $V_k$ .
- (3) Wenn  $w_p$  und  $w_q$  auf  $P_k$  nicht benachbart sind, so wird die Kante zwischen  $w_p$  und seinem Nachfolger auf  $P_k$  als erste horizontale Kante bezeichnet; die Kante zwischen  $w_q$  und seinem Vorgänger auf  $P_k$  wird als letzte horizontale Kante bezeichnet.
- (4) Wenn  $w_p$  und  $w_q$  auf  $P_k$  benachbart sind, wird im Folgenden das Einfügen von  $v_{k+1}$  als Typ-I-Einfügeoperation bezeichnet. Sind  $w_p$  und  $w_q$  nicht auf  $P_k$  benachbart, so wird das Einfügen als Typ-II-Einfügeoperation bezeichnet.



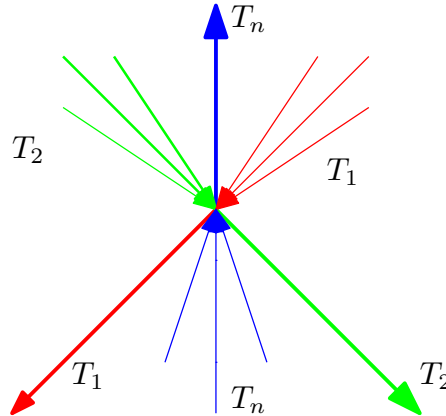
In [20] wurde nachgewiesen, dass jeder ebene Dreiecksgraph mit Außenfläche  $(v_1, v_2, v_n)$  eine kanonische Ordnung hat. Das Konzept der kanonischen Ordnung wurde dann auf dreifach zusammenhängende Graphen erweitert [27]. Hier ist—wie bei ebenen Dreiecksgraphen—die kanonische Ordnung im Allgemeinen nicht eindeutig bestimmt, so dass die Begriffe ganz linke kanonische Ordnung und ganz rechte kanonische Ordnung eingeführt worden sind [27][1]. Auf ebene Dreiecksgraphen übertragen ergibt sich:

**Definition 8:** Sei  $G = (V, E)$  ein ebener Dreiecksgraph mit  $|V| = n$  und der Außenfläche  $(v_1, v_2, v_n)$ . Eine kanonische Ordnung  $\sigma = (v_1^\sigma = v_1, v_2^\sigma = v_2, \dots, v_k^\sigma = v_k, \dots, v_n^\sigma = v_n)$  von  $G$  wird eine ganz linke kanonische Ordnung bzw. eine ganz rechte kanonische Ordnung genannt, wenn für jede weitere kanonische Ordnung  $\pi = (v_1^\pi = v_1, v_2^\pi = v_2, \dots, v_k^\pi = v_k, \dots, v_n^\pi = v_n)$  folgendes Kriterium erfüllt ist:

Sei  $r$  das kleinste Element aus der Menge  $\{1, 2, \dots, n\}$ , für das  $v_r^\sigma \neq v_r^\pi$  gilt. Sei  $v_p$  der Knoten mit  $v_r^\sigma = v_p$  und  $v_q$  der Knoten mit  $v_r^\pi = v_q$ . Wenn mit  $w_\ell$  der ganz linke Vorfahre von  $v_p$  in  $\sigma$  und mit  $w_{\ell'}$  der ganz linke Vorfahre von  $v_q$  in  $\pi$  bezeichnet wird, so gilt  $\ell < \ell'$  für eine ganz linke kanonische Ordnung beziehungsweise  $\ell > \ell'$  für eine ganz rechte kanonische Ordnung.

**Definition 9 ([40] in der Darstellung aus [45]):** Sei  $G$  ein ebener Dreiecksgraph und  $(v_1, v_2, v_n)$  die Außenfläche von  $G$  gegen den Uhrzeigersinn. Ein Schnyder-Realisierer  $R$  von  $G$  ist eine Aufteilung der inneren Kanten von  $G$  in drei Mengen  $T_1, T_2, T_n$  von gerichteten Kanten, so dass Folgendes gilt:

- (1) Für jedes  $i \in \{1, 2, n\}$  befinden sich die zu  $v_i$  inzidenten Kanten in  $T_i$  und sind gegen  $v_i$  gerichtet.
- (2) Für jeden inneren Knoten  $v$  von  $G$  hat  $v$  in jeder der drei Mengen  $T_1, T_2, T_n$  genau eine Kante, die  $v$  verlässt. Die Reihenfolge gegen den Uhrzeigersinn der zu  $v$  inzidenten Kanten lautet: ausgehend in  $T_1$ , eingehend in  $T_n$ , ausgehend in  $T_2$ , eingehend in  $T_1$ , ausgehend in  $T_n$  und eingehend in  $T_2$ . Jeder Abschnitt von eingehenden Kanten kann leer sein (Abbildung 9).



**Abbildung 9:** Schnyder-Realisierer, in Anlehnung an [6].

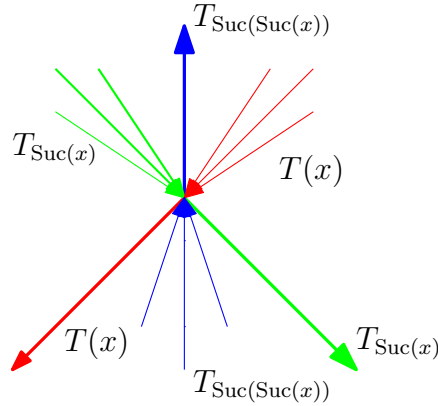
**Definition 10:** Die drei Mengen  $T_1, T_2, T_n$  stellen jeweils Bäume dar. Diese drei Bäume ergeben zusammen den Schnyder-Realisierer  $(T_1, T_2, T_n)$ . In [40] wird bewiesen, dass jeder ebene Dreiecksgraph mit Außenfläche  $(v_1, v_2, v_n)$  einen Schnyder-Realisierer hat. Ein Einzeler der drei Bäume wird im Folgenden auch ein Schnyder-Baum genannt; die zu  $T_1, T_2, T_n$  gehörenden Kanten werden mit  $T_1$ -Kanten,  $T_2$ -Kanten bzw.  $T_n$ -Kanten bezeichnet.

Eine steigende Kante soll eine mit einer positiven Steigung gezeichnete Kante sein; eine fallende Kante soll eine mit einer negativen Steigung gezeichnete Kante sein.

**Hilfssatz 3 ([6]):** Ein 3-Baum hat genau einen Schnyder-Realisierer.

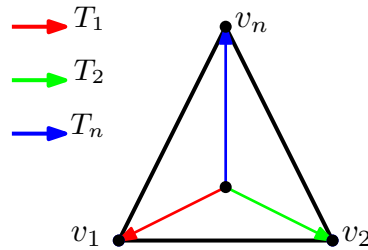
*Beweis.* Wegen der mit [6] nicht völlig vergleichbaren Terminologie—Schnyder-Realisierer gegenüber *3-orientation*—wird ein vollständiger Beweis von Hilfssatz 3 mittels Induktion durchgeführt.

Um die Reihenfolge der  $T_1$ -,  $T_2$ - und  $T_n$ -Kanten allgemeingültig notieren zu können, wird eine zyklische Nachfolgefunktion  $\text{Suc} : \{1, 2, n\} \rightarrow \{1, 2, n\}$  mit  $\text{Suc}(1) = 2$ ,  $\text{Suc}(2) = n$  und  $\text{Suc}(n) = 1$  definiert. Die Reihenfolge der Kanten in einem Schnyder-Realisierer ist in Abbildung 10 für  $x \in \{1, 2, n\}$  mit Hilfe dieser Nachfolgefunktion beschrieben.



**Abbildung 10:** Schnyder-Realisierer mit Nachfolgefunktion, in Anlehnung an [6].

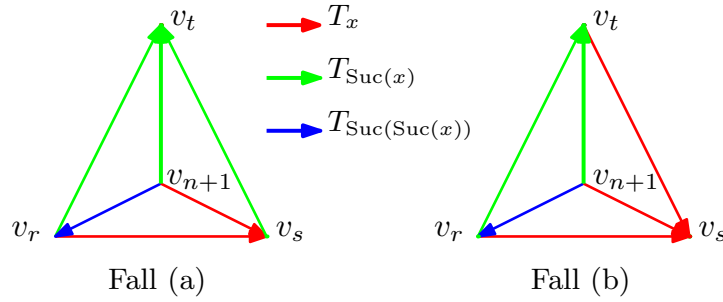
Basisfall: Für einen 3-Baum mit vier Knoten und der Außenfläche  $(v_1, v_2, v_n)$  ist der eineindeutige Schnyder-Realisierer in Abbildung 11 gezeigt.



**Abbildung 11:** Schnyder-Realisierer für den  $K_4$ .

Induktionsschluss: Sei für einen 3-Baum  $T$  mit  $n$  Knoten der eineindeutige Schnyder-Realisierer konstruiert. Der 3-Baum mit  $(n + 1)$  Knoten entsteht aus  $T$  durch eine Stapeloperation. Hierbei sei  $v_{n+1}$  auf der Fläche des Dreiecks mit den Eckpunkten  $v_r, v_s, v_t$

(Notierung gegen den Uhrzeigersinn) positioniert worden. Es wird konstruktiv nachgewiesen, dass der Realisierer für den neuen 3-Baum existiert und eineindeutig ist. Hierfür verbindet man  $v_{n+1}$  mit  $v_r, v_s, v_t$  derart, dass die drei neuen, von  $v_{n+1}$  ausgehenden Kanten den Bedingungen eines Schnyder-Realisierers entsprechen. Dies geschieht folgendermaßen: Die drei Knoten  $v_r, v_s, v_t$  werden hinsichtlich der Reihenfolge geordnet, in der sie in den Graphen eingefügt worden sind. Sei o.B.d.A.  $v_r$  der Knoten, der als letzter in den Graphen eingefügt worden ist. Aufgrund dieser Reihenfolge müssen  $(v_r, v_s)$  und  $(v_r, v_t)$  von  $v_r$  ausgehen. Die Konstruktion startet, indem man  $(v_r, v_s)$  als eine  $T_x$ -Kante mit  $x \in \{1, 2, n\}$  auffasst. Dann muss  $(v_r, v_t)$  aufgrund der Zugehörigkeit zu einem Schnyder-Realisierer eine  $T_{\text{Suc}(x)}$ -Kante sein. Dann wiederum muss  $(v_{n+1}, v_r)$  eine von  $v_{n+1}$  ausgehende  $T_{\text{Suc}(\text{Suc}(x))}$ -Kante sein. Es werden nun die beiden Fälle unterschieden (Abbildung 12):



**Abbildung 12:** Schnyder-Realisierer eines 3-Baums.

(a) Der Knoten  $v_s$  ist nach  $v_t$  eingefügt worden.

Dann geht  $(v_s, v_t)$  von  $v_s$  aus und ist eine  $T_{\text{Suc}(x)}$ -Kante. Die Kante  $(v_{n+1}, v_s)$  geht von  $v_{n+1}$  aus und kann dann nur eine  $T_x$ -Kante sein. Auch  $(v_{n+1}, v_t)$  geht von  $v_{n+1}$  aus und ist dann zwingend eine  $T_{\text{Suc}(x)}$ -Kante. Dies stimmt mit der erforderlichen Reihenfolge der Kanten um  $v_t$  überein.

(b) Der Knoten  $v_t$  ist nach  $v_s$  eingefügt worden.

Dann geht  $(v_t, v_s)$  von  $v_t$  aus und ist eine  $T_x$ -Kante. Die Kante  $(v_{n+1}, v_t)$  geht von  $v_{n+1}$  aus und kann dann nur eine  $T_{\text{Suc}(x)}$ -Kante sein. Auch  $(v_{n+1}, v_s)$  geht von  $v_{n+1}$  aus und ist dann zwingend eine  $T_x$ -Kante. Dies stimmt mit der erforderlichen Reihenfolge der Kanten um  $v_s$  überein.

In beiden Fällen war der Konstruktionsvorgang eineindeutig. Auch die Zuordnung zu den beiden Fällen war eineindeutig. Damit ist der Induktionsschluss bewiesen.

Mit Basisfall und Induktionsschluss ist auch die Behauptung bewiesen.  $\square$

**Hilfssatz 4 ([19]):** Eine kanonische Ordnung eines ebenen Dreiecksgraphen induziert eindeutig einen Schnyder-Realisierer. Mit den Bezeichnungen aus den Definitionen 5, 7 und 10 gilt für alle Knoten  $v_k$  von  $G$  hinsichtlich der zwischen  $v_k$  und  $G_{k-1}$  verlaufenden Kanten:

- (1) Die vom Knoten  $v_k$  zu seinem ganz linken Vorfahren in  $G_{k-1}$  verlaufende Kante gehört zu  $T_1$  und verlässt  $v_k$ .

- (2) Die vom Knoten  $v_k$  zu seinem ganz rechten Vorfahren in  $G_{k-1}$  verlaufende Kante gehört zu  $T_2$  und verlässt  $v_k$ .
- (3) Die anderen zu  $v_k$  inzidenten Kanten gehören zu  $T_n$  und sind auf  $v_k$  gerichtet.<sup>1</sup>

**Hilfssatz 5:** Die Kontur  $C_k$  besteht für  $3 \leq k \leq n$  nur aus  $T_1$ -Kanten und  $T_2$ -Kanten.

*Beweis.* Der Basisfall ist für  $k = 3$  erfüllt.

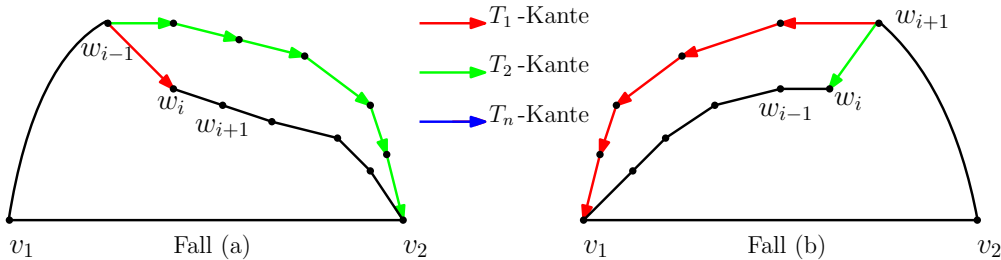
Induktionsschluss: Sei die Behauptung für  $k$  bewiesen. Ferner sei  $w_p$  der ganz linke Vorfahre bzw.  $w_q$  der ganz rechte Vorfahre von  $v_{k+1}$  in  $G_k$ . Die Kontur  $C_{k+1}$  erhält man durch Hinzufügen der ganz linken Kante und der ganz rechten Kante von  $v_{k+1}$  zur Kontur  $C_k = (w_1 = v_1, \dots, w_p, \dots, w_q, \dots, w_t = v_2)$ . Es folgt  $C_{k+1} = (w_1 = v_1, \dots, w_p, v_{k+1}, w_q, \dots, w_t = v_2)$ . Die beiden Abschnitte  $(w_1 = v_1, \dots, w_p)$  und  $(w_q, \dots, w_t = v_2)$  weisen laut Induktionsvoraussetzung keine  $T_n$ -Kanten auf. Da die ganz linke Kante bzw. die ganz rechte Kante nach Hilfssatz 4 eine  $T_1$ -Kante bzw. eine  $T_2$ -Kante ist, besteht auch die Kontur  $C_{k+1}$  nur aus  $T_1$ - und  $T_2$ -Kanten. Damit ist der Induktionsschluss bewiesen.

Mit Basisfall und Induktionsschluss ist auch die Behauptung bewiesen.  $\square$

**Hilfssatz 6:** Für die Kontur  $C_k = (w_1 = v_1, \dots, w_{i-1}, w_i, w_{i+1}, \dots, w_j = v_2)$  eines ebenen Dreiecksgraphen  $G = (V, E)$  gilt Folgendes: Wenn die Kante  $(w_{i-1}, w_i)$  eine  $T_1$ -Kante und die Kante  $(w_i, w_{i+1})$  eine  $T_2$ -Kante ist, kann  $w_i$  nicht Endpunkt einer Sehne sein.

*Beweis.* Zunächst wird bewiesen, dass  $w_{i-1}$  und  $w_{i+1}$  Vorfahren von  $w_i$  sind. Der Beweis wird indirekt geführt. Angenommen,  $w_{i-1}$  sei nicht Vorfahre von  $w_i$ . Dann ist  $w_i$  Vorfahre von  $w_{i-1}$ . In diesem Fall verläuft die  $T_1$ -Konturkante von  $w_{i-1}$  nach  $w_i$ . Bei  $w_{i-1}$  startet ferner im Schnyder-Baum  $T_2$  ein Weg nach  $v_2$ . Abbildung 13(a) verdeutlicht, dass in diesem Fall  $w_i$  nicht auf der Kontur liegen kann, was einen Widerspruch darstellt. Somit ist  $w_{i-1}$  Vorfahre von  $w_i$ .

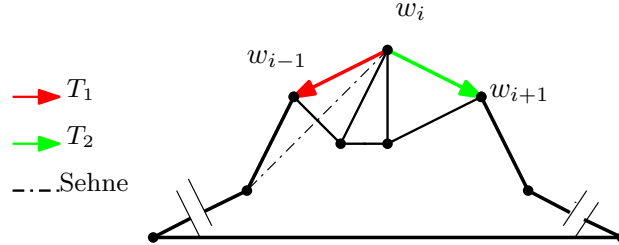
Nehme man nun für einen weiteren Widerspruchsbeweis an, dass  $w_{i+1}$  nicht Vorfahre von  $w_i$  ist. Dann ist  $w_i$  Vorfahre von  $w_{i+1}$ . In diesem Fall verläuft die  $T_2$ -Konturkante von  $w_{i+1}$  nach  $w_i$ . Bei  $w_{i+1}$  startet ferner im Schnyder-Baum  $T_1$  ein Weg nach  $v_1$ . Abbildung 13(b) verdeutlicht, dass in diesem Fall  $w_i$  nicht auf der Kontur liegen kann, was einen Widerspruch darstellt. Somit ist  $w_{i+1}$  Vorfahre von  $w_i$ .



**Abbildung 13:** (a) und (b): Erster Teil im Beweis von Hilfssatz 6.

<sup>1</sup>In [19] sind die Kanten eines Schnyder-Realisierers von der jeweiligen Wurzel weg orientiert, so dass dort die Kantenrichtung entgegengesetzt beschrieben wird.

In Verbindung mit Hilfssatz 4 ist  $w_{i-1}$  daher ganz linker Vorfahre und  $w_{i+1}$  ganz rechter Vorfahre von  $w_i$ . Dann existiert ein Weg  $P = (w_{i-1}, \dots, w_{i+1})$  zwischen  $w_{i-1}$  und  $w_{i+1}$ . Eine zu  $w_i$  inzidente, äußere Sehne kann es nicht geben, da sich dadurch der Verlauf der Kontur verändern würde. Eine zu  $w_i$  inzidente Sehne im Inneren von  $G_k$  kann es nicht geben, da die Sehne den Weg  $P$  kreuzen würde (Abbildung 14). Dies ist jedoch nicht möglich, da  $G$  ein ebener Dreiecksgraph ist.  $\square$



**Abbildung 14:** Ausgeschlossene Lage einer Sehne bezüglich der Kontur.

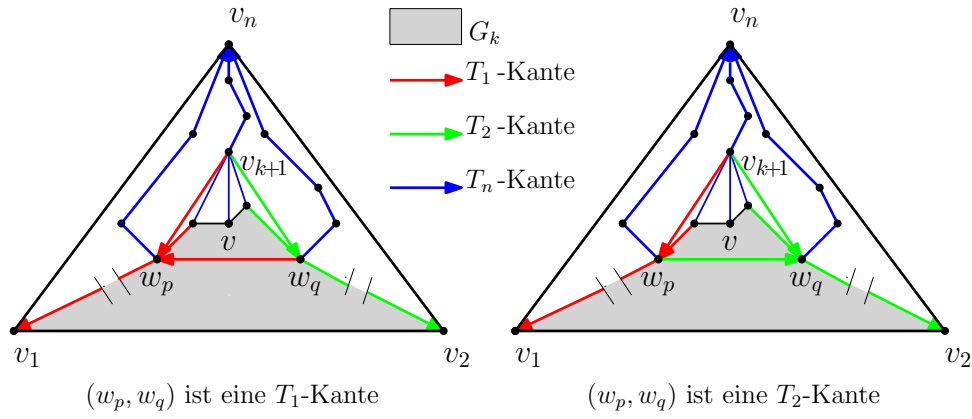
**Hilfssatz 7:**  $G$  sei ein ebener Dreiecksgraph mit Außenfläche  $(v_1, v_2, v_n)$  und ganz linker kanonischer Ordnung  $\sigma = (v_1, v_2, \dots, v_k, \dots, v_n)$ . Als letzter Knoten sei  $v_k$  in die Kontur  $C_k = (w_1 = v_1, \dots, w_{i-1}, w_i = v_k, w_{i+1} \dots, w_j = v_2)$  eingefügt worden. Der Abschnitt auf der aktuellen Kontur zwischen  $v_k$  und  $v_2$  werde mit  $C_k^* = (w_i = v_k, w_{i+1} \dots, w_j = v_2)$  bezeichnet. Dann wird  $C_k^*$  nur durch  $T_2$ -Kanten gebildet.

*Beweis.* Die kanonische Ordnung wird in umgekehrter Reihenfolge  $(v_n, v_{n-1}, \dots, v_3)$  durch sukzessives Entfernen geeigneter Knoten konstruiert. Ein Knoten ist dann geeignet, wenn er auf der aktuellen Kontur liegt und nicht Endpunkt einer Sehne dieser Kontur ist [20]. Wenn mehrere geeignete Knoten vorhanden sind, muss zur Konstruktion einer ganz linken kanonischen Ordnung von den in Frage kommenden Knoten immer der ganz rechte Knoten ausgewählt und entfernt werden. Andernfalls wären die Knoten später nicht im Sinne einer ganz linken kanonischen Ordnung geordnet.

Nach Hilfssatz 4 ist  $(w_i, w_{i+1})$  eine ganz rechte Kante und daher eine  $T_2$ -Kante. Damit muss der Beweis nur noch für den verbleibenden Bereich  $(w_{i+1}, \dots, w_j = v_2)$  geführt werden. Dies geschieht indirekt. Man nehme daher an, dass die Kontur  $C_k$  im Bereich  $(w_{i+1}, \dots, w_j = v_2)$  Kanten hat, die keine  $T_2$ -Kanten sind. Nach Hilfssatz 5 müssen dies  $T_1$ -Kanten sein. Mindestens eine dieser  $T_1$ -Kanten muss auf  $C_k$  in Richtung  $v_2$  eine  $T_2$ -Kante als Nachfolgerin haben, weil zumindest die allerletzte Kante auf  $C_k$  mit Endpunkt  $v_2$  eine  $T_2$ -Kante ist. Sei  $v_m$  der Knoten, an dem dieser Wechsel von einer  $T_1$ -Kante auf eine  $T_2$ -Kante stattfindet. Es muss  $m < k$  gelten, da  $v_k$  als letzter Punkt in die Kontur eingefügt worden ist. Nach Hilfssatz 6 sind sowohl  $v_k$  als auch  $v_m$  nicht inzident zu einer Sehne. Bei der Konstruktion der kanonischen Ordnung waren daher zu diesem Zeitpunkt sowohl  $v_m$  als auch  $v_k$  geeignete Knoten für die Entfernung. Wegen  $m < k$  muss hier  $v_k$  ausgewählt und entfernt worden sein. Von mehreren (hier zweien) geeigneten Kandidaten wurde also der ganz linke Knoten entfernt. Aufgrund der einleitenden Bemerkung zu diesem Beweis kann daher keine ganz linke kanonische Ordnung konstruiert worden sein. Dies ist ein Widerspruch. Damit ist die Behauptung bewiesen.  $\square$

**Hilfssatz 8:** Sei  $G = (V, E)$  ein 3-Baum mit kanonischer Ordnung  $\sigma$ . Mit den Bezeichnungen aus Definition 7 gilt:

- (a) Der ganz linke Vorfahre  $w_p$  und der ganz rechte Vorfahre  $w_q$  von  $v_{k+1}$  sind durch eine Kante miteinander verbunden.
- (b) Die Kante  $(w_p, w_q)$  ist entweder eine auf  $w_p$  gerichtete  $T_1$ -Kante oder eine von  $w_p$  ausgehende  $T_2$ -Kante.
- (c) Das entstandene Dreieck  $(w_p, w_q, v_{k+1})$  ist selbst wieder ein 3-Baum [33].
- (d) Sind  $w_p$  und  $w_q$  keine Nachbarn auf  $P_k$ , befinden sich die auf  $P_k$  zwischen  $w_p$  und  $w_q$  liegenden Knoten innerhalb des Dreiecks  $(w_p, w_q, v_{k+1})$ . Insbesondere stellt dann die erste horizontale Kante eine  $T_1$ -Kante, die letzte horizontale Kante eine  $T_2$ -Kante dar.



**Abbildung 15:** Existenz der Kante  $(w_p, w_q)$

*Beweis.* (a) Sind  $w_p$  und  $w_q$  Nachbarn auf  $P_k$ , so ist nichts weiter zu beweisen.

Im folgenden seien daher  $w_p$  und  $w_q$  keine Nachbarn auf  $P_k$  (Abbildung 15). Es muss daher mindestens einen Knoten  $v$  geben, der auf  $P_k$  zwischen  $w_p$  und  $w_q$  liegt. Die Knoten auf  $P_{k+1}$  stellen dann nach Abbildung 15 in  $G$  für  $X = \{v_n\}$  und  $Y = \{v\}$  einen  $X, Y$ -Separator dar. Sei dieser Separator mit  $S$  bezeichnet. Dann muss es einen minimalen  $X, Y$ -Separator  $S'$  mit  $S' \subseteq S$  geben. In  $S'$  sind somit nur Knoten des Wegs  $P_{k+1}$  enthalten. Da  $G$  ein 3-Baum ist, ist  $S'$  nach Hilfssatz 2 eine 3-Clique, also  $|S'| = 3$ . Sei mit  $G_{k+1}$  wie üblich der nach dem Hinzufügen von  $v_{k+1}$  entstandene Untergraph von  $G$  bezeichnet. Man betrachte nun im Schnyder-Baum  $T_n$  die drei Wege  $T_n^1 = (w_p, \dots, v_n)$ ,  $T_n^2 = (w_q, \dots, v_n)$ ,  $T_n^3 = (v_{k+1}, \dots, v_n)$ . Gemäß Hilfssatz 4(c) liegt der Zielknoten einer  $T_n$ -Kante in der kanonischen Ordnung nach dem Ursprungsknoten der Kante. Daher starten diese drei Wege zwar von  $P_{k+1}$ , verlaufen dann aber vollständig innerhalb von  $G \setminus G_{k+1}$ . Man betrachte nun die drei Wege  $W_1 = (v, \dots, w_p, \dots, v_n)$ ,  $W_2 = (v, \dots, w_q, \dots, v_n)$  und  $W_3 = (v, v_{k+1}, \dots, v_n)$ , wobei die beiden Teilwege  $W_1' = (v, \dots, w_p)$  und  $W_2' = (v, \dots, w_q)$  auf  $P_k$  liegen und die drei Teilwege  $W_1'' = (w_p, \dots, v_n)$ ,  $W_2'' = (w_q, \dots, v_n)$ ,  $W_3'' = (v_{k+1}, \dots, v_n)$  den zuvor beschriebenen Wegen in  $T_n$  entsprechen. Es gilt also  $W_1'' = T_n^1$ ,  $W_2'' = T_n^2$ ,  $W_3'' = T_n^3$ . Dann muss ein Knoten aus  $W_1$  (verschieden von  $v$  und  $v_n$ ) in  $S'$  liegen, da  $W_1$  sonst durch  $S'$  nicht separiert würde. Dieser Knoten kann nur  $w_p$  sein, da  $w_p$  der einzige Knoten von  $W_1$  ist, der

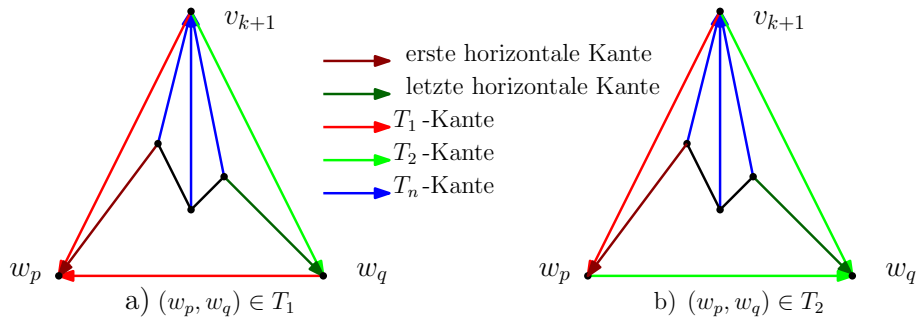
auf  $P_{k+1}$  liegt. Ausgehend von  $W_2$  bzw.  $W_3$  argumentiert man analog und erhält  $w_q \in S'$  bzw.  $v_{k+1} \in S'$ . Wegen  $|S'| = 3$  folgt  $S' = \{w_p, v_{k+1}, w_q\}$ . Da  $S'$  eine 3-Clique ist, folgt daraus unmittelbar die Existenz der Kante  $(w_p, w_q)$ .

Wenn  $w_p$  und  $w_q$  keine Nachbarn auf  $P_k$  sind, sind sie aufgrund des herbeigeführten Widerspruchs dennoch durch eine Kante miteinander verbunden. Damit ist die Behauptung bewiesen.<sup>2</sup>

(b) Wenn  $w_p$  und  $w_q$  Nachbarn auf  $P_k$  sind, ist  $(w_p, w_q)$  eine Kante auf der Kontur von  $G_k$ . Andernfalls stellt  $(w_p, w_q)$  eine Sehne dieser Kontur dar. Im ersten Fall muss sie nach Hilfssatz 5 eine  $T_1$ - oder  $T_2$ -Kante sein. Ist sie eine  $T_1$ -Kante, muss sie in Richtung auf  $v_2$  und damit auf  $w_q$  gerichtet sein. Ist sie eine  $T_2$ -Kante, muss sie in Richtung auf  $v_1$  und damit auf  $w_p$  gerichtet sein. Für den zweiten Fall nimmt man Bezug auf die Konstruktion von  $\sigma$ . Bei der Konstruktion von  $\sigma$  darf die Kante  $(w_p, w_q)$  erst entfernt werden, wenn sie keine Sehne mehr auf der gerade aktuellen Kontur ist [20]. Zum Zeitpunkt ihrer Entfernung ist sie daher Kante der gerade aktuellen Kontur. Wegen der Zugehörigkeit zu dieser Kontur kann sie nach Hilfssatz 5 auch im zweiten Fall nur eine  $T_1$ - oder  $T_2$ -Kante sein. Die ganz linke Kante  $(v_{k+1}, w_p)$  ist eine von  $v_{k+1}$  ausgehende  $T_1$ -Kante. Entsprechend ist  $(v_{k+1}, w_q)$  eine von  $v_{k+1}$  ausgehende  $T_2$ -Kante. Die Kante  $(w_p, w_q)$  kann daher wegen der Struktur eines Schnyder-Realisierers auch im zweiten Fall nur eine auf  $w_p$  gerichtete  $T_1$ -Kante oder eine von  $w_p$  ausgehende  $T_2$ -Kante (Abbildung 15) sein.<sup>3</sup>

(c) In Lemma 2 von [33] wurde bewiesen, dass ein beliebiges Dreieck in der Zeichnung eines 3-Baums selbst wieder die Zeichnung eines 3-Baums ist.

(d) Die Knoten auf  $P$  zwischen  $w_p$  und  $w_q$  müssen sich innerhalb dieses Dreiecks befinden. Andernfalls würden die  $T_n$ -Kanten, die einen dieser Knoten als ersten Endknoten und  $v_{k+1}$  als zweiten Endknoten haben, die Kante  $(w_p, w_q)$  kreuzen. Das betrachtete Dreieck selbst stellt nach Aussage (a) einen 3-Baum dar, es muss daher einen Schnyder-Realisierer induzieren. Da die drei Eckpunkte des Dreiecks die Wurzeln dieses Schnyder-Realisierers sind, folgt für die erste horizontale Kante und die letzte horizontale Kante die Behauptung aus den Gesetzmäßigkeiten eines Schnyder-Realisierers. Abbildung 16 zeigt, wie sich für  $(w_p, w_q) \in T_1$  und  $(w_p, w_q) \in T_2$  dasselbe Ergebnis ergibt.  $\square$



**Abbildung 16:** Erste und letzte horizontale Kante in einem Schnyder-Realisierer.

<sup>2</sup> $v_3$  und  $v_n$  stellen nicht erfasste Sonderfälle dar. Da  $v_1$  und  $v_2$  benachbart sind, bleibt der Hilfssatz auch für  $v_3$  und  $v_n$  gültig.

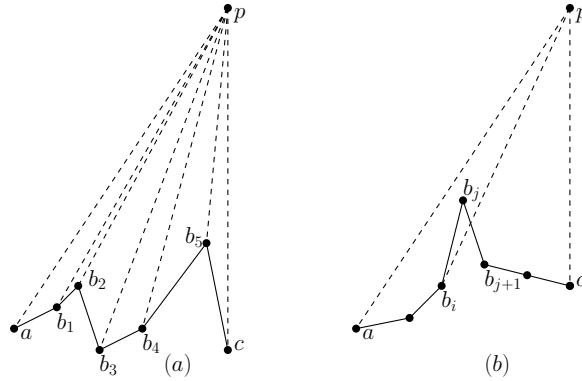
<sup>3</sup>Die zu  $v_{k+1} = v_3$  oder  $v_{k+1} = v_n$  gehörende Kante  $(v_1, v_2)$  gehört keinem Schnyder-Baum an.

Folgende Aussage über Sichtbarkeit von polygonalen Ketten ist bekannt:

**Hilfssatz 9 (Lemma 1 aus [14]):** Sei  $a, b_1, b_2, \dots, b_k, c (= b_{k+1})$  eine streng  $x$ -monotone polygonale Kette  $C$ . Sei  $p$  ein Punkt oberhalb von  $C$  derart, dass die Segmente  $ap$  und  $cp$  die polygonale Kette nur in  $a$  und  $c$  schneiden und sonst nicht. Wenn die Steigungen der steigenden Kanten (im Sinne von Definition 10) von  $C$  kleiner sind als die Steigung des Segmentes  $ap$  und die Steigungen der fallenden Kanten von  $C$  größer sind als die Steigung des Segmentes  $pc$ , dann ist jeder Punkt  $b_i$  sichtbar von  $p$  aus.

Für den später beschriebenen Algorithmus wird eine Variante dieses Hilfssatzes benötigt, bei der sich der Punkt  $p$  senkrecht oberhalb des rechten Endpunktes der polygonalen Kette befindet. In dieser Variante wird das Segment  $cp$  als fallend mit der Steigung  $-\infty$  angesehen. Die Steigungen der fallenden Kanten von  $C$  sind daher grundsätzlich größer als die Steigung des Segmentes  $cp$ . Man erhält so folgenden Hilfssatz:

**Hilfssatz 10 (Variante von Lemma 1 aus [14]):** Sei  $a, b_1, b_2, \dots, b_k, c (= b_{k+1})$  eine streng  $x$ -monotone polygonale Kette  $C$ . Sei  $p$  ein Punkt senkrecht oberhalb von  $c$  derart, dass die Segmente  $ap$  und  $cp$  die polygonale Kette nur in  $a$  und  $c$  schneiden und sonst nicht. Wenn die Steigungen der Kanten von  $C$  kleiner sind als die Steigung des Segmentes  $ap$ , dann ist jeder Punkt  $b_i$  sichtbar von  $p$  aus (Abbildung 17).



**Abbildung 17:** (a) Darstellung von Hilfssatz 10. (b) Darstellung des Beweises von Hilfssatz 10, jeweils nach [14].

*Beweis.* Er ist eine geringfügige Modifikation des Originalbeweises. Die dort verwendeten Begriffe *left edge* bzw. *right edge* stellen steigende Kanten bzw. fallende Kanten innerhalb einer streng  $x$ -monotonen polygonalen Kette dar. Für den Beweis der Variante kann die hervorgehobene Textstelle gestrichen werden, da sämtliche  $b_i$  mit  $1 \leq i \leq k$  sich nur in der linken Halbebene der vertikalen Linie durch  $P$  befinden können. Ansonsten bleibt der Beweis (Abbildung 17) gültig.

**Proof.** Suppose for a contradiction that some vertex  $b_i$ , where  $1 \leq i \leq k$ , is not visible to  $p$ . Without loss of generality assume that [Hervorhebung durch den Verfasser]  $b_i$  is in the left half-plane of the vertical line through  $p$ . Since  $C$  is strictly  $x$ -monotone, no left edge of  $C$  can block the visibility between  $p$  and  $b_i$ . Hence let  $(b_i, b_{i+1})$  be the right edge that blocks the visibility, where  $i < j \leq k$ , as shown in Figure 3(b). If the slope of the line  $l_{b_i, b_j}$  is smaller than  $\text{slope}(a, p)$ ,

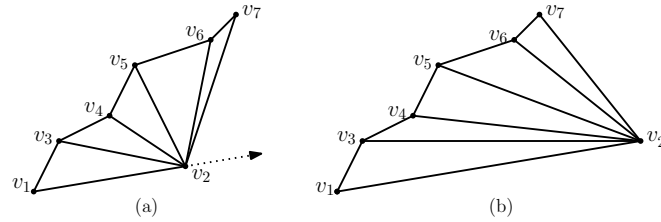


then it is also smaller than  $\text{slope}(p, b_i)$ , and hence  $b_i$  must be visible to  $p$ . We may thus assume that  $\text{slope}(b_i, b_j)$  is as large as  $\text{slope}(a, p)$ , which implies that  $(b_i, b_j)$  cannot be an edge in  $C$ . Consider now the path  $P = (b_i, b_{i+1} \dots, b_j)$ . Observe that the edge  $(b_i, b_{i+1})$  must lie to the right half plane of  $l_{b_i, b_j}$ . On the other hand, the edge  $(b_{j-1}, b_j)$  must lie to the left half plane of  $l_{b_i, b_j}$ . Hence there exists some edge  $e$  in  $P$  that crosses  $l_{b_i, b_j}$ . Since  $P$  is strictly  $x$ -monotone,  $e$  must be a left edge. Furthermore, since  $e$  crosses  $l_{b_i, b_j}$ , we have  $\text{slope}(e) > \text{slope}(b_i, b_j) \leq \text{slope}(a, p)$ , which is a contradiction. ([14], Seite 41)

□

**Hilfssatz 11:** Für  $G = (V, E)$  mit  $|V| = n$  bestehe  $E$  aus der sogenannten Grundlinie  $(v_1, v_2)$  sowie den Kanten  $(v_1, v_3)$ ,  $(v_k, v_{k+1})$ ,  $3 \leq k \leq n-1$  und  $(v_k, v_n)$ ,  $3 \leq k \leq n$ . In der Zeichnung  $\Gamma$  von  $G$  ist die Steigung jedes der Liniensegmente  $(v_1, v_3)$  bzw.  $(v_k, v_{k+1})$ ,  $1 \leq k \leq n-1$  größer als die Steigung der Grundlinie  $(v_1, v_2)$ . Die Grundlinie selbst ist mit einer Steigung größer oder gleich null gezeichnet.

Schiebt man  $v_2$  von  $v_1$  weg in Richtung der verlängerten Grundlinie, hat  $G$  in der hierdurch induzierten Zeichnung  $\Gamma'$  dieselbe Einbettung wie in  $\Gamma$  (Abbildung 18).



**Abbildung 18:** Verschiebung von  $v_2$  in Richtung der Grundlinie. (a) Vor der Verschiebung. (b) Nach der Verschiebung.

*Beweis.* Der verschobene Knoten werde mit  $v'_2$  bezeichnet.  $\Gamma'$  wird als Ergebnis eines iterativen Konstruktionsvorganges gedeutet. Hierbei werden nach Zeichnung der neuen Grundlinie  $(v_1, v'_2)$  sukzessive die Liniensegmente  $(v_1, v_3)$  und  $(v_k, v_{k+1})$ ,  $1 \leq k \leq n-1$  mit unveränderter Steigung gezeichnet. Die Folge dieser Teilzeichnungen werde mit  $\Gamma'_k$ ,  $3 \leq k \leq n$  bezeichnet. Induktiv wird bewiesen, dass in diesen Teilzeichnungen die Einbettung der Knoten dieselbe ist wie in  $\Gamma$ .

Der Basisfall besteht aus drei Knoten. Offenbar haben die Knoten  $(v_1, v'_2, v_3)$  nach der Verschiebung von  $v_2$  dieselbe Einbettung wie in  $(v_1, v_2, v_3)$ .

Induktionsschluss: Sei die Behauptung für  $k$  bewiesen und die Zeichnung  $\Gamma'_k$  erstellt. Nun wird  $v_{k+1}$  der Zeichnung hinzugefügt;  $(v_k, v_{k+1})$  und  $(v_{k+1}, v'_2)$  werden gezeichnet. Nach Voraussetzung ist die Steigung von  $(v_k, v_{k+1})$  größer als die der Grundlinie. Der Schnittpunkt der gedachten Verlängerung von  $(v_k, v_{k+1})$  mit der Grundlinie (bzw. deren Verlängerung) liegt daher so, dass nach der Verschiebung  $v'_2$  in derselben der beiden Halbebenen liegt, die durch  $(v_k, v_{k+1})$  induziert werden, wie zuvor  $v_2$  (Abbildung 19). Somit ändert sich die Orientierung des Dreiecks  $(v_k, v'_2, v_{k+1})$  nicht. Daher weisen die Knoten in  $\Gamma'_{k+1}$  dieselbe Einbettung auf wie in  $\Gamma$ . Der Induktionsschluss ist damit bewiesen.

Mit Basisfall und Induktionsschluss ist der Induktionsbeweis erbracht. Wegen  $\Gamma' = \Gamma'_n$  ist Hilfssatz 11 bewiesen. □

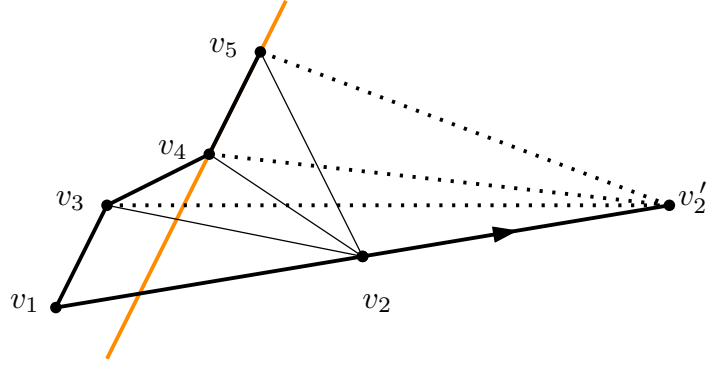


Abbildung 19: Darstellung des Beweises von Hilfssatz 11.

**Hilfssatz 12:** Für einen Graphen  $G = (V, E)$  mit aktueller Kontur  $C_k$  werde der Abschnitt der Kontur zwischen  $v_k$  und  $v_2$  mit  $C_k^*$  bezeichnet. Wird der Graph nach einer ganz linken kanonischen Ordnung konstruiert, liegt der ganz rechte Vorfahre von  $v_{k+1}$  auf dem Konturabschnitt  $C_k^*$ .

*Beweis.* Für einen Widerspruchsbeweis nehme man an, der ganz rechte Vorfahre von  $v_{k+1}$  liege nicht auf  $C_k^*$ . Nach Einfügen von  $v_{k+1}$  verläuft die Kontur im Bereich zwischen  $v_{k+1}$  und  $v_2$  dann unter anderem über die ganz rechte Kante von  $v_{k+1}$ , die ganz linke Kante von  $v_k$ ,  $v_k$  selbst und die ganz rechte Kante von  $v_k$ . Die ganz linke Kante von  $v_k$  ist daher Teil des Konturabschnitts  $C_{k+1}^*$ . Die ganz linke Kante von  $v_k$  ist jedoch eine  $T_1$ -Kante. Daher enthält  $C_{k+1}^*$  dann eine  $T_1$ -Kante (Abbildung 20). Da der Graph mit einer ganz linken kanonischen Ordnung konstruiert wurde, ist dies wegen Hilfssatz 7 ein Widerspruch. Damit ist bewiesen, dass der ganz rechte Vorfahre von  $v_{k+1}$  auf  $C_k^*$  liegt.  $\square$

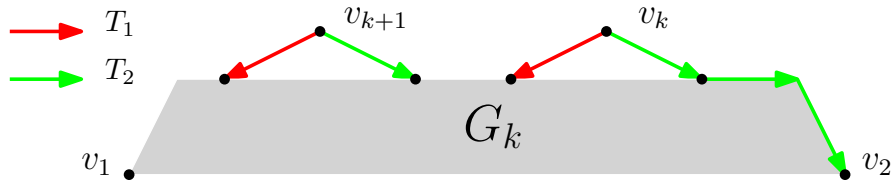


Abbildung 20: Darstellung des Beweises zu Hilfssatz 12.

### 2.2.2. Beschreibung von Algorithmus 3TREEFEWSEG

Die Strategie zur Reduzierung der Segmentanzahl folgt einer Idee aus [14]. Die Kanten eines der drei Bäume des Schnyder-Realisierers werden so gezeichnet, dass ein Segment, das an einem inneren Knoten dieses Baums beginnt, an einem Blatt des Baums endet. Die Knoten entlang des Pfads von einem inneren Knoten zu einem Blatt werden also mit kollinearen Punkten gezeichnet. Die Anzahl der Segmente eines so gezeichneten Baums ist dann gleich der Anzahl seiner Blätter. Die Gitterschranke wird durch eine Begrenzung der Steigung erreicht, mit der Kanten des betreffenden Schnyder-Baumes gezeichnet werden.

Der Algorithmus wird in Runden ausgeführt. Eine Runde besteht aus zwei Prozeduren. In der ersten Prozedur, OpInsert genannt, wird ein neuer Knoten auf dem Gitter positioniert. In der zweiten Prozedur, OpDiagonalShift bzw. OpDiagShift genannt, werden einzelne Knoten diagonal auf dem Gitter verschoben.

Der 3-Baum  $T$  sei in die Außenfläche  $(v_1, v_2, v_n)$  eingebettet. Vorbereitend wird zunächst ein Schnyder-Realisierer konstruiert. Dann wird der Baum bestimmt, der von den drei Schnyder-Bäumen die geringste Anzahl von Blättern hat. Ohne Beschränkung der Allgemeinheit sei  $T_1$  dieser Baum. Nun wird mit Bezug auf die Kante  $(v_1, v_2)$  eine ganz linke kanonische Ordnung von  $T$  berechnet. Jeder 3-Baum hat nach Hilfssatz 3 einen eindeutigen Schnyder-Realisierer. Daher stimmt der von der ganz linken kanonischen Ordnung induzierte Realisierer mit dem im ersten Schritt des Algorithmus konstruierten Realisierer überein. Fügt man während des Algorithmus die Knoten in der Reihenfolge der ganz linken kanonischen Ordnung in die Zeichnung ein, bleibt  $v_1$  somit Wurzel des Schnyder-Baums mit der geringsten Anzahl von Blättern. Ein von einer oder mehreren  $T_1$ -Kanten gebildetes gerades Liniensegment wird im folgenden als ein  $T_1$ -Segment bezeichnet.

Nach Hilfssatz 7 besteht die Kontur zwischen einem gerade eingefügten Knoten und  $v_2$  nur aus  $T_2$ -Kanten. Die Knoten auf diesem Konturabschnitt können daher einzeln in Verlängerung ihres  $T_1$ -Segmentes diagonal auf die Außenfläche hin verschoben werden.

Zum Nachhalten der für die Zeichnung der  $T_1$ -Segmente verwendeten Steigungen werden die Knoten in  $T_1$  mit zwei Attributen versehen. Ein Attribut, Ausgangslabel genannt, enthält die Steigung der ausgehenden  $T_1$ -Kante; das andere Attribut, Eingangslabel genannt, beinhaltet die größte Steigung der bisher gezeichneten eingehenden  $T_1$ -Kanten. Im Eingangslabel zeigt der voreingestellte Wert null an, dass bisher für den Knoten keine eingehende Kante gezeichnet worden ist. Ein globaler Zähler speichert die letzte überhaupt vergebene Steigung für eine  $T_1$ -Kante.

Im Einzelnen lautet die Abfolge des Algorithmus:

1. OpInsert: Der nächste einzufügende Knoten  $v_{k+1}$  ist durch die ganz linke kanonische Ordnung eindeutig bestimmt. Der ganz linke Vorfahre von  $v_{k+1}$  ist Elternknoten von  $v_{k+1}$  in  $T_1$ ; durch eine Abfrage in  $T_1$  kann man diesen Elternknoten ermitteln.

Das Eingangslabel des Elternknoten zeigt an, ob  $v_{k+1}$  der erste Kindknoten von ihm ist, der gezeichnet werden soll. In diesem Fall wird der ganz linken Kante von  $v_{k+1}$  durch Auslesen des Ausgangslabels die Steigung der vom Elternknoten ausgehenden  $T_1$ -Kante zugewiesen. Andernfalls wird differenziert, ob es sich um eine Typ-I-Einfügeoperation oder um eine Typ-II-Einfügeoperation handelt. Wenn beim Einfügen von  $v_{k+1}$  der ganz linke und der ganz rechte Vorfahre von  $v_{k+1}$  auf der Kontur benachbart sind, spricht man von einer Typ-I-Einfügeoperation, sonst von einer Typ-II-Einfügeoperation. Im ersten Fall wird das Eingangslabel um eins inkrementiert und die so ermittelte Steigung der ganz linken Kante von  $v_{k+1}$  zugewiesen. Sollte diese Steigung die größte der bisher vergebenen Steigungen sein, muss der globale Zähler ebenfalls noch inkrementiert werden. Bei einer Typ-II-Einfügeoperation wird grundsätzlich der globale Zähler inkrementiert und dieser neue Wert der Steigung der ganz linken Kante von  $v_{k+1}$  zugewiesen.

Bei einem solchen Vorgehen induziert—wie gewünscht—jedes Blatt von  $T_1$  genau ein Segment. Die Begründung für die Differenzierung zwischen einer Typ-I- und Typ-II-Einfügeoperation erfolgt ab Seite 41 im Rahmen des Beweises.

Mit einer Abfrage in  $T_2$  wird der Elternknoten in  $T_2$  und somit der ganz rechte Vorfahre von  $v_{k+1}$  bestimmt. Die  $x, y$ -Koordinaten bereits gezeichneter Knoten werden in einem Attribut des Knotens gespeichert. Da der ganz linke und der ganz rechte Vorfahre von  $v_{k+1}$  bereits verarbeitet wurden, können ihre  $x, y$ -Koordinaten diesem Attribut entnommen werden. Der Knoten  $v_{k+1}$  selbst wird nun so gezeichnet, dass seine  $x$ -Koordinate mit der  $x$ -Koordinate seines ganz rechten Vorfahren übereinstimmt.

Aus den  $x, y$ -Koordinaten seines ganz linken Vorfahren, der zugewiesenen Steigung für seine ganz linke Kante und der ihm zugewiesenen  $x$ -Koordinate ist die Position von  $v_{k+1}$  auf dem Gitter eindeutig bestimmt. Es können nun seine ganz linke Kante, der Knoten selbst und seine ganz rechte Kante (grundsätzlich eine senkrecht fallende Kante) gezeichnet werden. Falls auf der Kontur von  $G_k$  weitere Knoten zwischen ganz linken Vorfahren und ganz rechten Vorfahren von  $v_{k+1}$  liegen, werden noch die  $T_n$ -Kanten zwischen  $v_{k+1}$  und diesen Knoten gezeichnet. OpInsert ist nun beendet, es beginnt OpDiagShift.

2. OpDiagShift: Wie schon erwähnt, werden während OpDiagShift die Knoten auf der

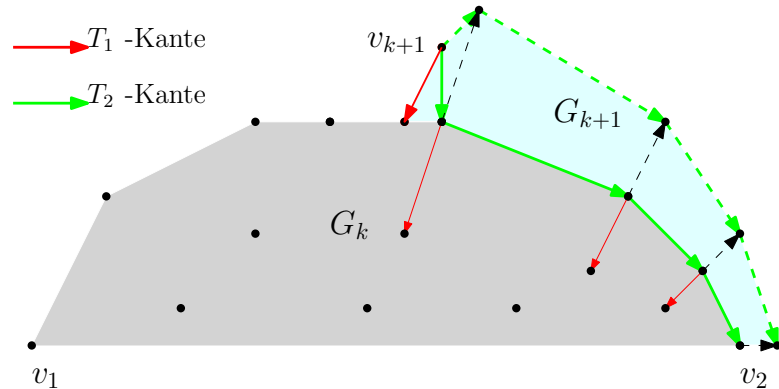


Abbildung 21: OpDiagShift im Überblick.

Kontur zwischen  $v_{k+1}$  und  $v_2$  sukzessive diagonal auf die Außenfläche hin verschoben, und zwar in Verlängerung ihrer bereits gezeichneten  $T_1$ -Kante. Diese Verschiebung beginnt mit dem Knoten folgend auf  $v_{k+1}$  und endet mit  $v_2$  (Abbildung 21). Den Knoten  $v_2$  verschiebt man dabei in Richtung der  $x$ -Achse. Das Maß der  $x$ -Komponente der diagonalen Verschiebung beträgt dabei für jeden der betroffenen Knoten eine Längeneinheit. Nach Verschiebung von  $v_2$  ist OpDiagShift und damit auch eine Runde beendet.

### 2.2.3. Eigenschaften von Algorithmus 3TREEFEWSEG

Für die Darstellung werden einige Bezeichnungen definiert:

- (1) Der Elternknoten von  $v_i$  in  $T_1$  wird mit  $\text{par}_i$  bezeichnet.
- (2) Als  $\text{lca}_{i,j}$  wird der niedrigste gemeinsame Vorfahre von zwei Knoten  $v_i, v_j$  in  $T_1$  bezeichnet.

- (3) Es werde bei einer Typ-II-Einfügeoperation der Knoten  $v_{k+1}$  eingefügt. Dann bezeichnet man die Gesamtheit der Knoten und Kanten auf dem Weg von dem ganz linken Vorfahren zu dem ganz rechten Vorfahren als polygonale Kette der (direkten) Vorfahren von  $v_{k+1}$ .
- (4) Die auf einen Knoten  $v_i$  gerichteten  $T_n$ -Kanten werden durch die ganz linke Kante und die ganz rechte Kante von  $v_i$  eingerahmt. Es wird im Folgenden von der (eindeutig bestimmten) rahmenden linken Kante bzw. rahmenden rechten Kante einer  $T_n$ -Kante gesprochen.
- (5) Eine  $T_1$ -Kante und  $T_2$ -Kante sind zueinander korrespondierend, wenn beide von demselben Knoten ausgehen.

**Satz 2:** Sei  $T = (V, E)$  ein 3-Baum mit ganz linker kanonischer Ordnung  $\sigma$ .

Es bedeutet (Abbildung 22):

- (1)  $\Gamma_k$  die Zeichnung nach der  $k$ -ten Runde des Algorithmus 3TREEFEWSEG;
- (2)  $\xi_k$  die Anzahl der gezeichneten Blätter im Schnyder-Baum  $T_1$  nach der  $k$ -ten Runde;
- (3)  $s_k$  die Anzahl der Segmente, mit denen  $T_1$  nach der  $k$ -ten Runde gezeichnet ist;
- (4)  $C_k = (w_1 = v_1, \dots, w_m = v_k, \dots, w_u = v_2)$  die aktuelle Kontur;
- (5)  $C_k^* = (w_m = v_k, \dots, w_u = v_2)$  der Abschnitt der aktuellen Kontur zwischen  $v_k$  und  $v_2$ ;
- (6)  $\text{out}_i^1$  die Steigung der vom Punkt  $v_i$  ( $1 \leq i \leq k$ ) ausgehenden  $T_1$ -Kante  
( $\text{out}_i^1$  wird auch Steigung des  $T_1$ -Segments von  $v_i$  genannt);
- (7)  $\text{out}_i^2$  die Steigung der vom Punkt  $v_i$  ( $1 \leq i \leq k$ ) ausgehenden  $T_2$ -Kante;
- (8)  $\text{range}_k$  der Wertebereich der Steigungen, mit denen die  $T_1$ -Segmente gezeichnet werden.
- (9)  $\eta_k$  die größte Steigung in  $T_1$  nach der  $k$ -ten Runde.

Dann verfügt  $\Gamma_k$  für  $3 \leq k \leq n$  über die folgenden neun Eigenschaften E 1 bis E 9:

- (E 1)** Die Knoten auf  $C_k$  stellen eine streng  $x$ -monotone polygonale Kette dar. Dabei sind die Knoten auf  $C_k^*$  in  $x$ -Richtung jeweils genau eine Längeneinheit voneinander entfernt.
- (E 2)** Die  $T_1$ -Kanten sind mit insgesamt  $s_k = \xi_k$  unterschiedlichen  $T_1$ -Segmenten gezeichnet. Die Steigungen sind ganzzahlig mit  $\text{range}_k = (1, \eta_k)$  und  $1 \leq \eta_k \leq \xi_k$ .
- (E 3)** Für einen Knoten  $v_i \neq v_2$  auf  $C_k^*$  sei  $v_j$  der Nachfolger von  $v_i$  in Richtung  $v_2$ . Dann gilt:
- (a)  $\text{lca}_{i,j} = \text{par}_j$ .  
Das abgeschlossene Gebiet, das durch den Weg  $(v_i, \dots, \text{par}_j)$ , die Kante  $(\text{par}_j, v_j)$  und die Konturkante  $(v_i, v_j)$  begrenzt wird, ist daher wohldefiniert. Es wird Domäne der beiden Knoten  $v_i$  und  $v_j$  genannt und mit  $\text{dom}_{i,j}$  bezeichnet. Für die Domäne gilt:
  - (b) Jede  $T_1$ -Kante der Domäne hat eine Steigung, die größer oder gleich der Steigung von Kante  $(\text{par}_j, v_j)$  ist. Dabei besteht Gleichheit nur für  $(\text{par}_j, v_j)$  selbst. Für eine allgemeingültige Darstellung wird zu diesem Zweck  $\text{out}_2^1 = 0$  definiert, was mit der Verschiebung von  $v_2$  in waagerechter Richtung während OpDiagShift konsistent ist.
- (E 4)** Für den Knoten  $v_i \in C_k$  einer ausgehenden  $T_2$ -Konturkante ist  $\text{out}_i^1 > \text{out}_i^2$ .
- (E 5)** Die Steigung einer  $T_n$ -Kante ist größer als die Steigung ihrer rahmenden linken Kante.

(E 6) Sei  $v_i$  ein Knoten auf  $C_k^*$ , der durch eine Typ-II-Operation eingefügt wurde. Dann gilt: Die Kanten in der polygonalen Kette der Vorfahren von  $v_i$  wurden seit seinem Einfügen mit Ausnahme der letzten horizontalen Kante nicht mehr verändert.

(E 7) Sei  $v_i$  ein Knoten auf  $C_k^*$ , der durch eine Typ-II-Operation eingefügt wurde. Dann gilt: Die Steigung der letzten horizontalen Kante in der polygonalen Kette seiner Vorfahren ist grundsätzlich kleiner als die ihrer korrespondierenden  $T_1$ -Kante.

(E 8) Die Zeichnung  $\Gamma_k$  ist kreuzungsfrei. Darüber hinaus wurde die Einbettung eines Knotens nach seinem Einfügen nicht mehr verändert.

(E 9) Der Knoten  $v_1$  hat die Koordinaten  $(0,0)$ ,  $v_2$  hat die Koordinaten  $(k-1,0)$ .

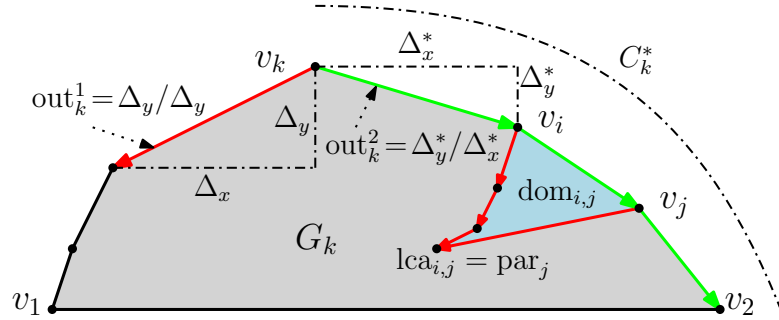


Abbildung 22: Verwendete Definitionen in der Beschreibung von Algorithmus 3TREEFEWSEG.

#### 2.2.4. Beweis von Algorithmus 3TREEFEWSEG:

Um zu beweisen, dass Algorithmus 3TREEFEWSEG korrekt ist, muss die Gültigkeit von Satz 2 nachgewiesen werden. Hierfür muss gezeigt werden, dass die Eigenschaften E 1 bis E 9 für  $3 \leq k \leq n$  gültig in  $\Gamma_k$  sind. Dies geschieht induktiv. Es wird zunächst der Basisfall und anschließend der Induktionsschluss für E 1 bis E 9 bewiesen.

**Basisfall:** Da die Zeichnung  $\Gamma_3$  mit  $v_1 = (0,0)$ ,  $v_2 = (2,0)$ ,  $v_3 = (1,1)$  offensichtlich die Eigenschaften E 1 bis E 9 erfüllt, ist der Beweis des Basisfalls erbracht.

**Induktionsschluss:** Sei die Gültigkeit von E 1 bis E 9 in  $\Gamma_k$  bewiesen. Auf den folgenden Seiten wird einzeln gezeigt, dass unter dieser Induktionsvoraussetzung die Eigenschaften E 1 bis E 9 auch in  $\Gamma_{k+1}$  gültig sind. Mit  $v_\ell$  wird während des gesamten Induktionsschlusses der ganz linke Vorfahre und mit  $v_r$  der ganz rechte Vorfahre von  $v_{k+1}$  notiert.

**Eigenschaft E 1:** Nach Hilfssatz 12 in Verbindung mit der Konstruktionsvorschrift wird  $v_{k+1}$  genau senkrecht über einem auf  $C_k^*$  liegenden Knoten eingefügt. Die nach Induktionsvoraussetzung genau eine Längeneinheit voneinander entfernt liegenden Knoten auf  $C_k^*$  werden in OpDiagShift—beginnend mit dem Nachfolger von  $v_{k+1}$ —sukzessive bis einschließlich  $v_2$  diagonal verschoben. Dabei beträgt die  $x$ -Komponente der Verschiebung genau eine Längeneinheit. Am Ende einer Runde stellt die Kontur  $C_{k+1}$  daher wieder eine streng  $x$ -monotone polygonale Kette dar. Auch liegen die Knoten auf  $C_{k+1}^*$  in  $x$ -Richtung jeweils eine Längeneinheit voneinander entfernt. Die Zeichnung  $\Gamma_{k+1}$  erfüllt somit E 1.

**Eigenschaft E 2:** Um Anzahl und Steigungen der  $T_1$ -Segmente nach der  $(k+1)$ -ten Runde zu berechnen, wird für OpInsert folgende Fallunterscheidung getroffen:

(a) Wird  $v_{k+1}$  in OpInsert als erstes Kind seines Elternknoten in  $T_1$  gezeichnet, kann  $v_{k+1}$  in Verlängerung eines bestehenden  $T_1$ -Segmentes gezeichnet werden. Die Anzahl der gezeichneten Blätter und der  $T_1$ -Segmente erhöhen sich daher nicht:

$$\xi_{k+1} = \xi_k \quad (1)$$

$$s_{k+1} = s_k. \quad (2)$$

Daraus folgt mit der Induktionsvoraussetzung  $s_k = \xi_k$  und Gleichung 1:

$$s_{k+1} = s_k = \xi_k = \xi_{k+1} \text{ beziehungsweise} \quad (3)$$

$$s_{k+1} = \xi_{k+1}. \quad (4)$$

Die Anzahl der (unterschiedlichen) Steigungen erhöht sich nicht:

$$\eta_{k+1} = \eta_k. \quad (5)$$

Zusammen mit der Induktionsvoraussetzung  $\eta_k \leq \xi_k$  und Gleichung 1 folgt hieraus für den Wertebereich der Steigungen:

$$\text{range}_{k+1} = (1, \eta_{k+1}) \text{ mit } 1 \leq \eta_{k+1} = \eta_k \leq \xi_k = \xi_{k+1} \text{ bzw.} \quad (6)$$

$$\text{range}_{k+1} = (1, \eta_{k+1}) \text{ mit } 1 \leq \eta_{k+1} \leq \xi_{k+1}. \quad (7)$$

(b) Wird  $v_{k+1}$  in OpInsert nicht als erstes Kind seines Elternknoten in  $T_1$  gezeichnet, erhöht sich die Anzahl der Blätter in  $T_1$  um eins; auch wird nach Konstruktionsvorschrift ein neues  $T_1$ -Segment erzeugt:

$$\xi_{k+1} = \xi_k + 1 \quad (8)$$

$$s_{k+1} = s_k + 1. \quad (9)$$

Daraus folgt mit der Induktionsvoraussetzung  $s_k = \xi_k$  und Gleichung 8:

$$s_{k+1} = s_k + 1 = \xi_k + 1 = \xi_{k+1} \text{ beziehungsweise} \quad (10)$$

$$s_{k+1} = \xi_{k+1}. \quad (11)$$

Die Anzahl der (unterschiedlichen) Steigungen erhöht sich höchstens um eins:

$$\eta_{k+1} \leq \eta_k + 1. \quad (12)$$

Zusammen mit der Induktionsvoraussetzung  $\eta_k \leq \xi_k$  und Gleichung 8 folgt hieraus für den Wertebereich der Steigungen:

$$\text{range}_{k+1} = (1, \eta_{k+1}) \text{ mit } 1 \leq \eta_{k+1} \leq \eta_k + 1 \leq \xi_k + 1 = \xi_{k+1} \text{ bzw.} \quad (13)$$

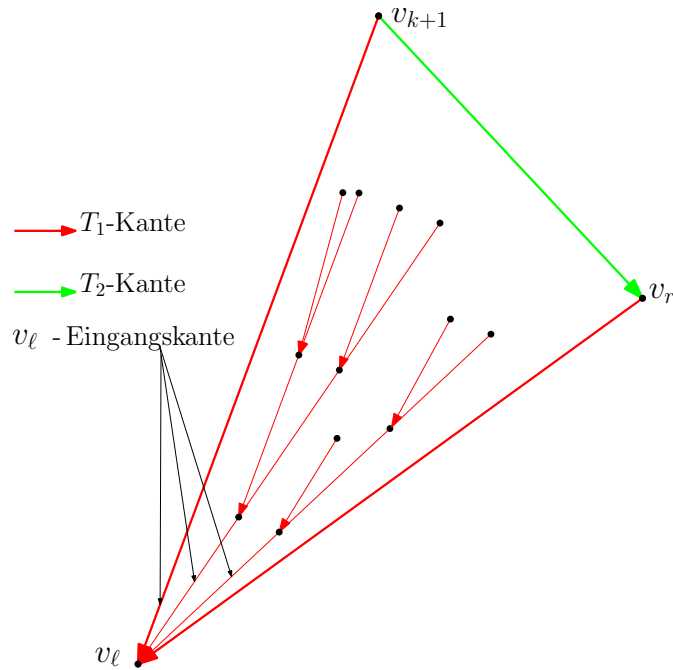
$$\text{range}_{k+1} = (1, \eta_{k+1}) \text{ mit } 1 \leq \eta_{k+1} \leq \xi_{k+1}. \quad (14)$$

Während OpDiagShift werden die Knoten auf  $C_k^*$  in Verlängerung ihres  $T_1$ -Segmentes verschoben. Dadurch verändert sich in (a) und (b) weder die Anzahl der  $T_1$ -Segmente noch deren Steigung bzw. der Wertebereich der Steigungen. Die Gleichungen (4) und (7) bzw. (11) und (14) beschreiben daher den Sachverhalt nach der  $(k+1)$ -ten Runde. Die Anzahl

der Segmente beträgt  $s_{k+1} = \xi_{k+1}$ . Der Wertebereich der Steigungen beträgt  $(1, \eta_{k+1})$  mit  $1 \leq \eta_{k+1} \leq \xi_{k+1}$ . Folglich ist E 2 gültig in  $\Gamma_{k+1}$ .

**Eigenschaft E 3:** Sei  $v_i$  ein Knoten auf  $C_k^*$  und  $v_j$  sein Nachfolger auf der Kontur in Richtung  $v_2$ . Dann besagt E 3, dass jede  $T_1$ -Kante in  $\text{dom}_{i,j}$  eine größere Steigung als die Kante  $(\text{par}_j, v_j)$  hat. Um im Induktionsschluss die Gültigkeit von E 3 auch für  $v_{k+1}$  nachzuweisen, muss gezeigt werden, dass in der Domäne von  $v_{k+1}$  und  $v_r$  jede  $T_1$ -Kante eine größere Steigung als die Kante  $(\text{par}_r, v_r)$  hat. Nach Hilfssatz 8(a) existiert die Kante  $(v_\ell, v_r)$  zwischen ganz linkem und ganz rechten Vorfahren von  $v_{k+1}$ . Entsprechend Hilfssatz 8(b) unterscheidet man zwei Fälle:

(a) Die Kante  $(v_\ell, v_r)$  ist eine  $v_r$  verlassende  $T_1$ -Kante. Es folgt direkt  $\text{lca}_{k+1,r} = v_\ell = \text{par}_r$ . In Abbildung 23 ist dieser Sachverhalt veranschaulicht.

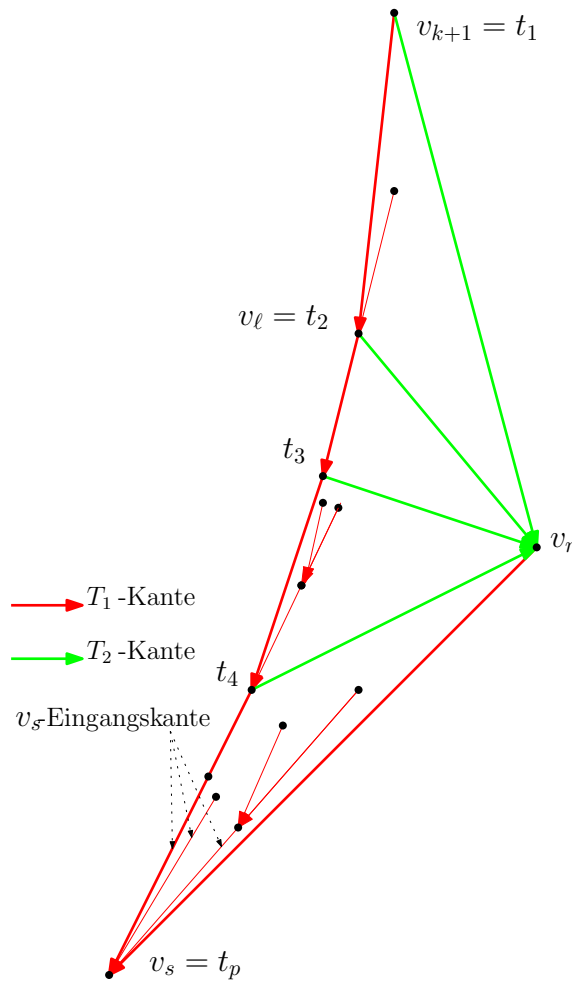


**Abbildung 23:** Erster Fall: Domäne von  $v_{k+1}$  und  $v_r$  mit  $(v_\ell, v_r) \in T_1$ .

Die resultierende Domäne ist in diesem Fall das Dreieck  $(v_\ell, v_r, v_{k+1})$ , das selbst wieder nach Hilfssatz 8(c) einen 3-Baum darstellt. Daher muss  $v_\ell$  Wurzel eines Schnyder-Baums für diesen 3-Baum sein. Aufgrund der Inzidenz von  $v_\ell$  zu den  $T_1$ -Kanten  $(v_{k+1}, v_\ell)$  und  $(v_\ell, v_r)$  kann dies nur der Schnyder-Baum  $T_1$  sein. Sämtliche  $T_1$ -Kanten der Domäne sind somit Kanten eines  $T_1$ -Weges von einem Knoten der Domäne über  $v_\ell$  nach  $v_1$ . Man betrachte nun jeweils die auf  $v_\ell$  gerichtete Kante eines solchen  $T_1$ -Weges. Sie sei  $v_\ell$ -Eingangskante des Weges genannt. Da konstruktionsbedingt die Steigungen der einzelnen Kanten eines  $T_1$ -Weges zur Wurzel  $v_1$  hin nicht zunehmen können, muss jede  $T_1$ -Kante der Domäne mindestens eine so große Steigung aufweisen wie die  $v_\ell$ -Eingangskante des zugehörigen Weges. Da die  $v_\ell$ -Eingangskante jedes aus der Domäne herausführenden  $T_1$ -Weges eine größere Steigung als  $(v_\ell, v_r)$  hat, muss dann auch jede  $T_1$ -Kante der Domäne eine größere Steigung als  $(v_\ell, v_r) = (\text{par}_r, v_r)$  haben. Im Fall (a) ist E 3 also gültig für  $v_{k+1}$ .



(b) Die Kante  $(v_\ell, v_r)$  ist eine auf  $v_r$  gerichtete  $T_2$ -Kante (Abbildung 24).



**Abbildung 24:** Zweiter Fall: Domäne von  $v_{k+1}$  und  $v_r$  mit  $(v_\ell, v_r) \in T_2$ .

Der Knoten  $v_r$  ist in diesem Fall der ganz rechte Vorfahre von  $v_\ell$ . Mit Hilfssatz 8(a) ist der ganz linke Vorfahre von  $v_\ell$  dann durch eine Kante mit  $v_r$  verbunden. Diese Kante kann nach Hilfssatz 8(b) eine  $v_r$  verlassende  $T_1$ -Kante oder eine auf  $v_r$  gerichtete  $T_2$ -Kante sein. Bei der zweiten Möglichkeit lässt sich die vorangegangene Überlegung erneut anwenden. Dies motiviert zur folgenden rekursiven Argumentation:

Man betrachte den  $T_1$ -Weg von  $v_\ell$  nach  $v_1$ . Solange die Knoten auf diesem Weg durch eine  $T_2$ -Kante mit  $v_r$  verbunden sind, ist auch der nächste sich auf diesem Weg befindliche Knoten mit  $v_r$  durch eine Kante verbunden. Sei  $v_s$  der erste Knoten auf diesem Weg, für den diese Verbindungskante keine  $T_2$ -Kante ist. Nach Hilfssatz 8(b) ist diese Kante dann eine auf  $v_s$  gerichtete  $T_1$ -Kante. Einen solchen Knoten  $v_s$  gibt es, weil spätestens die Wurzel  $v_1$  keine  $T_2$ -Kante aufweist.

Offensichtlich ist  $\text{par}_r = v_s$  und es gibt einen  $T_1$ -Weg von  $v_{k+1}$  über  $v_\ell$  nach  $v_s$ . Daraus folgt  $\text{lca}_{(k+1),r} = \text{par}_r$ . Die zu betrachtende Domäne ist in diesem Fall durch den  $T_1$ -Weg  $(v_{k+1}, \dots, v_s)$ , die Kante  $(v_s, v_r)$  und die Konturkante  $(v_{k+1}, v_r)$  begrenzt. Der Weg  $(v_{k+1}, \dots, v_s)$  werde in  $(t_1 = v_{k+1}, t_2 = v_\ell, \dots, t_p = v_s)$  umbenannt. Jede  $T_1$ -Kante der

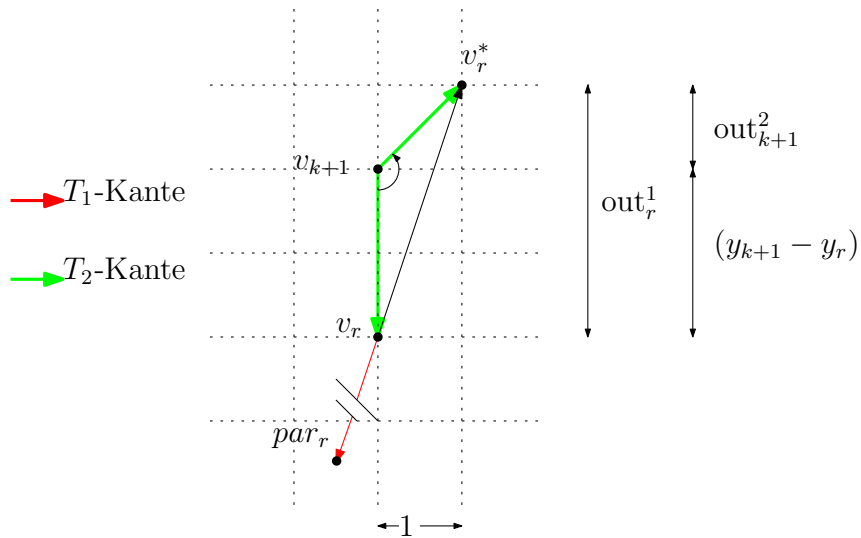
Domäne muss eine Kante von einem der induzierten 3-Bäume  $(t_i, t_{i+1}, v_r)$ ,  $1 \leq i \leq p-1$  sein. Analog zum ersten Fall argumentiert man, dass eine solche Kante Teil eines  $T_1$ -Wegs von einem Knoten der Domäne über  $t_{i+1}$  und  $v_s$  (gegebenenfalls  $t_{i+1} = v_s$ ) nach  $v_1$  ist. Man betrachte die auf  $v_s$  gerichtete Kante eines solchen Wegs. Analog zu Fall (a) sei sie  $v_s$ -Eingangskante des Weges genannt. Die  $v_s$ -Eingangskante jedes aus der Domäne herausführenden Weges hat eine größere Steigung als  $(v_s, v_r)$ . Analog zum ersten Fall zeigt man dann, dass jede  $T_1$ -Kante der Domäne eine größere Steigung hat als  $(v_s, v_r) = (\text{par}_r, v_r)$ . Auch im Fall (b) ist E 3 daher gültig für  $v_{k+1}$ .

Beide Fälle zusammen ergeben die Gültigkeit von E 3 für  $v_{k+1}$ . Nach Hilfssatz 12 befand sich  $v_r$  auch schon auf  $C_k^*$ . Nach Induktionsvoraussetzung ist E 3 daher auch gültig für die Knoten auf der Kontur von  $v_r$  bis  $v_2$ . In  $\Gamma_{k+1}$  gilt also E 3.

**Eigenschaft E 4:** Die Eigenschaft 4 besagt, dass für den Knoten  $v_i$  einer ausgehenden  $T_2$ -Konturkante  $\text{out}_i^1 > \text{out}_i^2$  ist. Im folgenden Induktionsschluss wird  $\Gamma_{k+1}$  betrachtet und nachgewiesen, dass E 4 auch in  $\Gamma_{k+1}$  gültig ist. Hinsichtlich der Lage von  $v_i$  zum gerade eingefügten Knoten  $v_{k+1}$  werden drei Fälle unterschieden:

(a) Für  $v_i$  auf  $C_{k+1}$  zwischen  $v_1$  und  $v_{k+1}$  gilt nach Induktionsvoraussetzung zu Beginn der  $(k+1)$ -ten Runde  $\text{out}_i^1 > \text{out}_i^2$ . Weder die von  $v_i$  ausgehende  $T_2$ -Konturkante noch ihre korrespondierende  $T_1$ -Kante werden in der  $(k+1)$ -ten Runde verändert. Die Relation  $\text{out}_i^1 > \text{out}_i^2$  ist im Fall (a) daher auch in  $\Gamma_{k+1}$  gültig.

(b) Nun sei  $v_i = v_{k+1}$ . Beide Typen einer Einfügeoperation fügen die neue  $T_2$ -Konturkante  $(v_{k+1}, v_r)$  als senkrecht fallende Kante ein. Die  $y$ -Koordinaten der beiden Endpunkte seien mit  $y_{k+1}$  bzw. mit  $y_r$  notiert. Durch OpDiagShift wird  $v_r$  diagonal mit der Steigung  $\text{out}_r^1$  auf die Außenfläche hin verschoben. Der verschobene Knoten werde mit  $v_r^*$  bezeichnet. Das Maß für die horizontale Komponente der Verschiebung beträgt eine Längeneinheit. Daraus folgt (Abbildung 25):



**Abbildung 25:** Verschiebung von  $(v_{k+1}, v_r)$  im Fall  $v_i = v_{k+1}$ .  
Beispielhaft mit  $\text{out}_r^1 = 3$ .

$$\text{out}_{k+1}^2 = \text{out}_r^1 - (y_{k+1} - y_r) \quad (15)$$

$$\Rightarrow \text{out}_r^1 > \text{out}_{k+1}^2 \text{ wegen } y_{k+1} - y_r > 0. \quad (16)$$

Im Induktionsschluss zu E 3 (Seite 34) wurde gezeigt, dass die Kante  $(v_\ell, v_{k+1})$  in der Domäne von  $v_{k+1}$  und  $v_r$  liegt und eine größere Steigung als  $(\text{par}_r, v_r)$  hat.<sup>4</sup> In anderer Notation bedeutet dies:

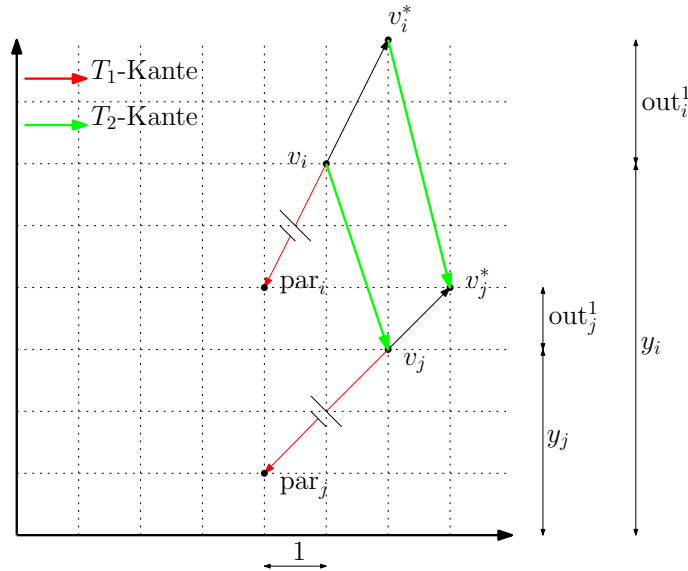
$$\text{out}_r^1 < \text{out}_{k+1}^1 \quad (17)$$

$$\Rightarrow \text{out}_{k+1}^2 < \text{out}_{k+1}^1 \text{ mit Gleichung 16.} \quad (18)$$

Die Gleichung 18 bedeutet, dass E 4 für Fall (b) Gültigkeit in  $\Gamma_{k+1}$  hat.

(c) Nun liege  $v_i$  auf  $C_{k+1}^*$  mit  $v_i \neq v_{k+1}$  und  $v_i \neq v_2$ . Nach Hilfssatz 12 lag  $v_i$  bereits auf  $C_k^*$ , so dass nach Induktionsvoraussetzung zu Beginn der  $(k+1)$ -ten Runde  $\text{out}_i^1 > \text{out}_i^2$  gilt. Es wird gezeigt, dass diese Relation auch in  $\Gamma_{k+1}$  gültig bleibt—also auch nach Verschiebung des Punktes  $v_i$  in OpDiagShift während der  $(k+1)$ -ten Runde.

Sei  $v_j$  der Nachfolger von  $v_i$  auf  $C_{k+1}^*$  in Richtung  $v_2$ . Die Steigung der Kante  $(v_i, v_j)$  in  $\Gamma_k$  und in  $\Gamma_{k+1}$  wird nun unterschiedlich bezeichnet: in  $\Gamma_k$  wie gewohnt mit  $\text{out}_i^2$  und in  $\Gamma_{k+1}$  nun mit  $\text{out}_i^{2*}$ . Die Werte  $\text{out}_i^1$  und  $\text{out}_j^1$  ändern sich entsprechend der Konstruktion beim Übergang von  $\Gamma_k$  nach  $\Gamma_{k+1}$  nicht. Mit  $y_i$  und  $y_j$  werden die  $y$ -Koordinaten von  $v_i$  bzw.  $v_j$  bezeichnet. Die verschobenen Knoten werden mit  $v_i^*$  und  $v_j^*$  notiert. Letztendlich sei in  $\Gamma_{k+1}$  die  $y$ -Koordinate von  $v_i^*$  mit  $y_i^*$  bezeichnet, die von  $v_j^*$  mit  $y_j^*$ . Art und Maß der Verschiebung stimmen mit Fall (b) überein (Abbildung 26). Dann beträgt:



**Abbildung 26:** Verschiebung von  $(v_i, v_j)$  im Fall  $v_i \in C_{k+1}^*$ .  
Beispielhaft mit  $\text{out}_i^1 = 2, \text{out}_j^1 = 1$ .

<sup>4</sup>Es besteht die methodische Schwierigkeit, dass die Induktion in Runden verläuft, eine Runde jedoch aus zwei abgegrenzten Prozeduren besteht. Der Zwischenzustand wird daher durch die Induktion nicht modelliert. Ein eigentlich gewünschter Bezug auf den Zwischenzustand muss somit durch den Bezug auf einen geeigneten, zuvor getätigten Induktionsschluss ersetzt werden.

$$\text{out}_i^{2*} = y_j^* - y_i^* \quad (19)$$

$$\text{out}_i^{2*} = (y_j + \text{out}_j^1) - (y_i + \text{out}_i^1) \quad (20)$$

$$\text{out}_i^{2*} = (y_j - y_i) - (\text{out}_i^1 - \text{out}_j^1) \quad (21)$$

$$\text{out}_i^{2*} = \text{out}_i^2 - (\text{out}_i^1 - \text{out}_j^1). \quad (22)$$

Die Knoten  $v_i, v_j$  lagen bereits auf  $C_k^*$  (Hilfssatz 12). Nach Induktionsvoraussetzung E 3 liegt daher  $(\text{par}_i, v_i)$  in  $\text{dom}_{i,j}$ . Wiederum mit Induktionsvoraussetzung E 3 ist die Steigung von  $(\text{par}_i, v_i)$  somit größer als die von  $(\text{par}_j, v_j)$ . In anderer Notation bedeutet dies:

$$\text{out}_i^1 > \text{out}_j^1 \quad \text{und mit Gleichung (22)} \quad (23)$$

$$\text{out}_i^{2*} < \text{out}_i^2. \quad (24)$$

Wie einleitend erwähnt gilt:

$$\text{out}_i^2 < \text{out}_i^1 \quad \text{und mit Gleichung (24)} \quad (25)$$

$$\text{out}_i^{2*} < \text{out}_i^1. \quad (26)$$

Die Gleichung 26 bedeutet, dass E 4 für Fall (c) Gültigkeit in  $\Gamma_{k+1}$  hat.

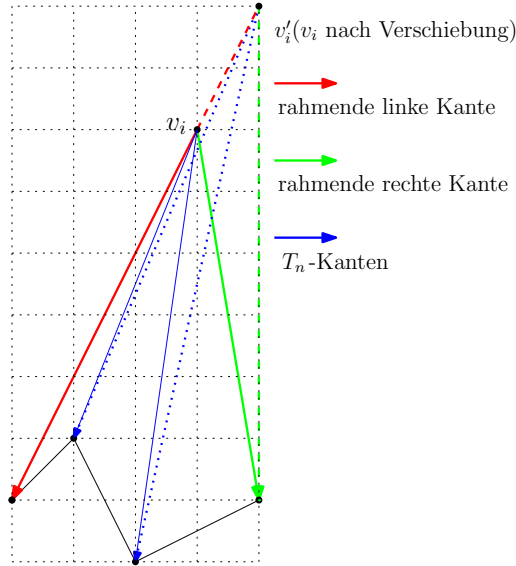
Mit diesen drei Fällen zusammen ist die Gültigkeit von E 4 in  $\Gamma_{k+1}$  für alle möglichen Lagen einer  $T_2$ -Konturkante bewiesen.

**Eigenschaft E 5:** Diese Eigenschaft besagt, dass die Steigung einer  $T_n$ -Kante in  $\Gamma_k$  größer ist als die Steigung ihrer rahmenden linken Kante. Um für den Induktionsschluss zu zeigen, dass diese Aussage auch in  $\Gamma_{k+1}$  gültig ist, wird folgende Fallunterscheidung durchgeführt:

- (1) Auf die zu Beginn der  $(k+1)$ -Runde bereits existierenden  $T_n$ -Kanten kann die Induktionsvoraussetzung E 5 angewendet werden. Sie haben daher vor Beginn von OpInsert eine größere Steigung als ihre rahmende Kante. Da diese Kanten in der  $(k+1)$ -ten Runde durch OpInsert in ihrer Lage nicht verändert werden, haben sie auch vor Beginn von OpDiagShift eine größere Steigung als ihre rahmende linke Kante.
- (2) In der  $(k+1)$ -ten Runde werden im Fall einer Typ-II-Einfügeoperation auch neue  $T_n$ -Kanten erzeugt. Diese sind zu  $v_{k+1}$  inzident und befinden sich im Bereich zwischen ganz linker Kante und ganz rechter Kante von  $v_{k+1}$ . Da die ganz rechte Kante durch OpInsert als senkrechte Kante eingefügt wird, sind die Steigungen der neu eingefügten  $T_n$ -Kanten größer als die Steigung der ganz linken Kante von  $v_{k+1}$ . Die ganz linke Kante von  $v_{k+1}$  stellt die rahmende linke Kante der betrachteten  $T_n$ -Kanten dar. Somit haben auch alle neu eingefügten  $T_n$ -Kanten nach OpInsert eine größere Steigung als ihre rahmende linke Kante.

Des Weiteren muss gezeigt werden, dass in beiden Fällen diese Aussage auch nach OpDiagShift noch wahr ist. Dies kann für die beiden Fälle in Einem erfolgen. Hierfür wird untersucht, ob die  $T_n$ -Kanten und rahmende linke Kante während OpDiagShift ihre Steigung ändern. Der grundsätzliche Ablauf des Algorithmus schließt eine Änderung der Steigung der

rahmenden linken Kante aus. Es bleibt also die Betrachtung der  $T_n$ -Kanten. Abbildung 27 zeigt, wie OpDiagShift die Steigung von  $T_n$ -Kanten vermindert. Sie sind jedoch weiterhin steigende Kanten und ihre Steigung ist weiterhin größer als die ihrer rahmenden linken Kante. Zusammengefasst ist E 5 also auch am Ende der  $(k + 1)$ -ten Runde und daher in  $\Gamma_{k+1}$  gültig.



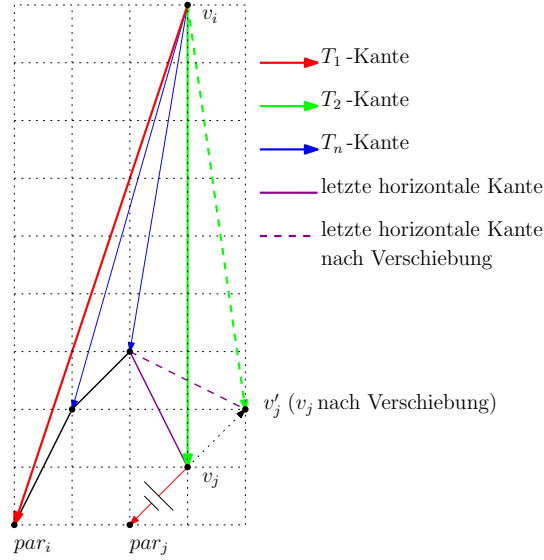
**Abbildung 27:** Veränderung der Steigung der  $T_n$ -Kanten während OpDiagShift.

**Eigenschaft E 6:** Sei  $v_i$  ein Knoten, der auf  $C_k^*$  liegt und durch eine Typ-II-Operation eingefügt wurde. Die Eigenschaft E 6 besagt, dass die Kanten in der polygonalen Kette seiner Vorfahren seit seinem Einfügen mit Ausnahme der letzten horizontalen Kante nicht mehr verändert wurden. Für den Induktionsschluss muss gezeigt werden, dass diese Aussage auch für einen Knoten  $v_i \in C_{k+1}^*$  gültig ist, der durch eine Typ-II-Operation eingefügt wurde. Folgende Fallunterscheidung wird getroffen:

- 1() Der Knoten  $v_i$  ist  $v_{k+1}$  selbst.  
Dann entsteht die polygonale Kette seiner Vorfahren erst in der  $(k + 1)$ -ten Runde während OpInsert.
- 2() Der Knoten  $v_i$  ist ungleich  $v_{k+1}$ .  
Nach Hilfssatz 12 lag  $v_i$  dann bereits auf  $C_k^*$ , so dass für ihn die Induktionsvoraussetzung gilt. Die polygonale Kette seiner Vorfahren ist daher mit Ausnahme der letzten Kante bis zum Ende der  $k$ -ten Runde nicht verändert worden. Diese Aussage bleibt auch während OpInsert gültig, da in OpInsert die polygonale Kette überhaupt nicht geändert wird.

Für beide Fälle ist noch nachzuweisen, dass die polygonale Kette der Vorfahren von  $v_i$  mit Ausnahme der letzten horizontalen Kante in der  $(k + 1)$ -ten Runde auch während OpDiagShift nicht verändert wird. Dies kann für die beiden Fälle in Einem geschehen. Sei  $v_j$  der Nachfolger von  $v_i$  auf  $C_{k+1}^*$  in Richtung  $v_2$ . Betrachtet wird nun die Verschiebung von  $v_j$

während OpDiagShift. Nach Abbildung 28 tritt dabei eine Änderung in der polygonalen Kette der Vorfahren von  $v_i$  auf. Diese Änderung betrifft jedoch nur die letzte horizontale Kante. Von der Änderung sind die anderen Kanten der polygonalen Kette nicht betroffen, was zu beweisen war. Daraus folgt die Gültigkeit von E 6 in  $\Gamma_{k+1}$ .



**Abbildung 28:** Verschiebung der letzten horizontalen Kante.

**Eigenschaft E 7:** Sei  $v_i$  ein Knoten, der auf  $C_k^*$  liegt und durch eine Typ-II-Operation eingefügt wurde. Die Eigenschaft E 7 besagt, dass in  $\Gamma_k$  die Steigung der letzten horizontalen Kante in der polygonalen Kette seiner Vorfahren —im Folgenden als die betrachtete letzte horizontale Kante bezeichnet—grundsätzlich kleiner ist als die ihrer korrespondierenden  $T_1$ -Kante. Für den Induktionsschluss muss gezeigt werden, dass diese Aussage auch für einen Knoten  $v_i \in C_{k+1}^*$  gültig ist, der durch eine Typ-II-Operation eingefügt wurde. Die betrachtete Aussage ist wohlgeformt, da die letzte horizontale Kante nach Hilfssatz 8(d) grundsätzlich eine  $T_2$ -Kante und ihre korrespondierende Kante daher eine  $T_1$ -Kante ist. Folgende Fallunterscheidung wird getroffen:

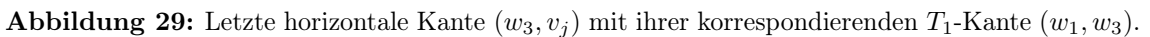
- (1) Der Knoten  $v_i$  ist  $v_{k+1}$  selbst.

Dann war die betrachtete letzte horizontale Kante in  $\Gamma_k$  die von  $v_r$  in Richtung  $v_1$  verlaufende Konturkante. Da die letzte horizontale Kante eine  $T_2$ -Kante (Hilfssatz 8(d)) ist und sie in  $\Gamma_k$  auf der Kontur lag, war sie in  $\Gamma_k$  eine  $T_2$ -Konturkante. Daher hatte sie in  $\Gamma_k$  nach E 4 eine kleinere Steigung als ihre korrespondierende  $T_1$ -Kante. Da sich in OpInsert die Steigungen existierender Kanten nicht ändern, gilt diese Aussage auch nach dem Einfügen von  $v_{k+1}$  unmittelbar vor Beginn von OpDiagShift.

- (2) Der Knoten  $v_i$  ist ungleich  $v_{k+1}$ .

Dann war die betrachtete letzte horizontale Kante auch schon in  $\Gamma_k$  die letzte horizontale Kante in dieser polygonalen Kette. Somit kann auf diese Kante die Induktionsvoraussetzung angewendet werden. Vor Beginn von OpInsert war daher ihre Steigung kleiner als die Steigung ihrer korrespondierenden Kante. Dies gilt ebenfalls für den Zeitpunkt unmittelbar vor OpDiagShift, da in OpInsert Kantensteigungen nicht verändert wurden.

Sei  $v_j$  wieder der Nachfolger von  $v_i$  auf  $C_{k+1}^*$  in Richtung  $v_2$ . Die Änderung in der Steigung der betrachteten letzten horizontalen Kante tritt bei der Verschiebung von  $v_j$  während OpDiagShift auf. Man betrachte nun die zu dieser Kante korrespondierende  $T_1$ -Kante. Sie liegt in der Domäne von  $v_i$  und  $v_j$ . Die Steigung dieser  $T_1$ -Kante ist nach E 3 größer als die Steigung der Kante  $(par_j, v_j)$ . Entsprechend liegt  $v_j$  vor und nach seiner Verschiebung in derselben der beiden Halbebenen, die durch die Verlängerung der korrespondierende  $T_1$ -Kante induziert werden. Die Steigung der betrachteten letzten horizontalen Kante muss daher auch nach der Verschiebung von  $v_j$  kleiner sein als die Steigung ihrer korrespondierenden  $T_1$ -Kante, was zu beweisen war. Die Eigenschaft E 7 ist daher gültig in  $\Gamma_{k+1}$ .



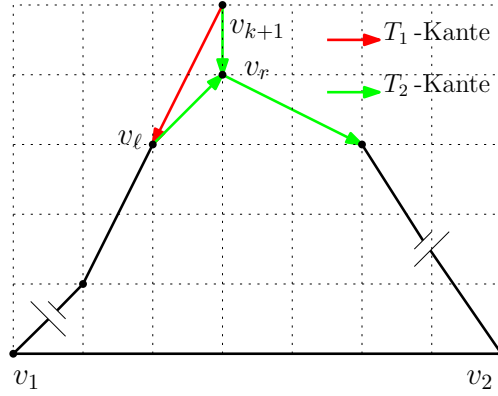
41

zweiten Schritt wird nachgewiesen, dass dies auch nach OpDiagShift zutrifft. Im letzten Schritt wird aufgezeigt, dass in der  $(k+1)$ -Runde die Einbettung bereits eingefügter Knoten bewahrt worden ist. Die einzelnen Schritte verlaufen folgendermaßen:

**(1) Kreuzungsfreiheit der Kanten in OpInsert:**

Es wird gezeigt, dass in OpInsert der Knoten  $v_{k+1}$  so auf der Außenfläche von  $G_k$  eingefügt wird, dass keine kreuzenden Kanten erzeugt werden. Aus Hilfssatz 8 ist bekannt, dass  $v_\ell$  und  $v_r$  durch die Kante  $(v_\ell, v_r)$  miteinander verbunden sind und dass diese Kante entweder eine auf  $v_\ell$  gerichtete  $T_1$ -Kante oder eine von  $v_\ell$  ausgehende  $T_2$ -Kante ist. Die weitere Darstellung orientiert sich an der Entscheidungsfindung für die Zuweisung der Steigungen von  $T_1$ -Segmenten (vgl. Seite 29). Man unterscheidet nun:

(a) Der Knoten  $v_{k+1}$  ist der erste Kindknoten von  $v_\ell$  in  $T_1$ , der gezeichnet wird. Die Kante  $(v_\ell, v_r)$  ist dann die von  $v_\ell$  ausgehende  $T_2$ -Kante (Abbildung 30).



**Abbildung 30:** OpInsert im Detail, Fall (a).

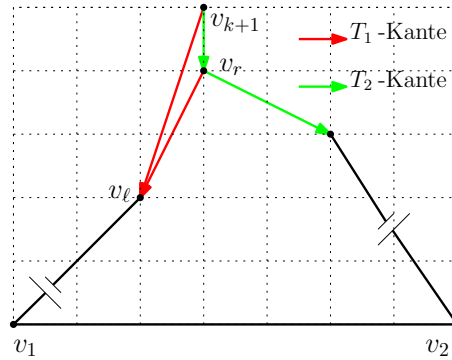
Es wird nun gezeigt, dass es sich in diesem Fall um eine Typ-I-Einfügeoperation handeln muss und  $(v_\ell, v_r)$  daher eine Konturkante ist. Zum Zweck eines Widerspruchsbeweises nehme man an, es würde sich um eine Typ-II-Einfügeoperation handeln. Dann wäre  $(v_\ell, v_r)$  keine Konturkante. Die Knoten auf  $C_k$  zwischen  $v_\ell$  und  $v_r$  würden in  $\Gamma_{k+1}$  innerhalb des Dreiecks  $(v_\ell, v_r, v_{k+1})$  liegen und die auf  $v_\ell$  gerichtete erste horizontale Kante würde eine  $T_1$ -Kante darstellen (Hilfssatz 8 in Verbindung mit Abbildung 16). Dann aber wäre  $v_{k+1}$  nicht mehr der erste Kindknoten von  $v_\ell$ , der gezeichnet wird, was einen Widerspruch bedeutet. Somit ist bewiesen, dass es sich um eine Typ-I-Einfügeoperation handelt.

Die Kontur  $C_k$  stellt laut E 1 eine streng  $x$ -monotone polygonale Kette dar. Da eine Typ-I-Einfügeoperation vorliegt, ist  $v_r$  der unmittelbare Nachfolger von  $v_\ell$  auf dieser Kette. Der Knoten  $v_{k+1}$  kann daher dann auf der Außenfläche von  $G_k$  ohne Kantenkreuzungen positioniert werden, wenn die Steigung der ganz linken Kante von  $v_{k+1}$  größer ist als die Steigung der Kante  $(v_\ell, v_r)$ . Die ganz linke Kante von  $v_{k+1}$  wird mit der Steigung des  $T_1$ -Segmentes von  $v_\ell$  gezeichnet, also mit  $\text{out}_\ell^1$ . Die Steigung der Kante  $(v_\ell, v_r)$  ist gleich  $\text{out}_\ell^2$ . Wegen E 4 gilt  $\text{out}_\ell^1 > \text{out}_\ell^2$ . Da genau diese Relation zu zeigen war, kann  $v_{k+1}$  im Fall (a) ohne Kantenkreuzungen auf der Außenfläche positioniert werden.



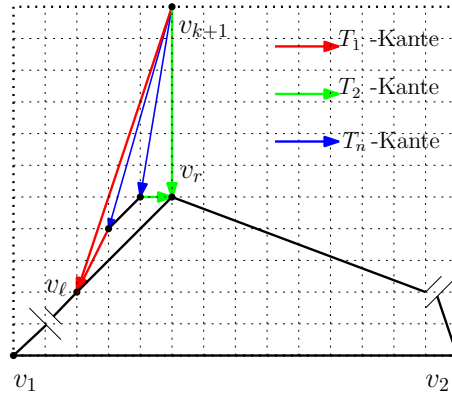
(b) Der Knoten  $v_{k+1}$  ist nicht der erste Kindknoten von  $v_\ell$  in  $T_1$ , der gezeichnet wird. Die Kante  $(v_\ell, v_r)$  ist dann die von  $v_r$  ausgehende  $T_1$ -Kante. In diesem Fall differenziert man weiter im Hinblick auf den Typ der Einfügeoperation:

(b.1) Es handelt sich um eine Typ-I-Einfügeoperation (Abbildung 31). Die Steigung der ganz linken Kante von  $v_{k+1}$  erhält dann den um eins inkrementierten Wert der Steigung von  $(v_\ell, v_r)$ . Die Situation ist nun analog zu (a): Die Steigung von  $(v_\ell, v_{k+1})$  ist größer als die von  $(v_\ell, v_r)$ . Da die anderen Bedingungen ebenfalls analog zu (a) sind, kann  $v_{k+1}$  auch im Fall (b.1) ohne Kantenkreuzungen auf der Außenfläche positioniert werden.



**Abbildung 31:** OpInsert im Detail, Fall (b.1).

(b.2) Es handelt sich um eine Typ-II-Einfügeoperation (Abbildung 32). In diesem Fall liegt die Kante  $(v_\ell, v_r)$  nicht auf der Kontur.



**Abbildung 32:** OpInsert im Detail, Fall (b.2).

Die Steigung der ganz linken Kante von  $v_{k+1}$  erhält dann den um eins inkrementierten Wert des globalen Zählers. Diese Kante wird daher mit der bisher größten Steigung einer  $T_1$ -Kante gezeichnet. Auf der Kontur  $C_k$  können nach Hilfssatz 5 keine  $T_n$ -Kanten liegen. Ferner hat jede auf  $C_k$  liegende  $T_2$ -Kante nach E 4 eine geringere Steigung als ihre korrespondierende  $T_1$ -Kante. Die ganz linke Kante von  $v_{k+1}$  hat daher eine größere Steigung als alle Konturkanten von  $C_k$ . Dies gewährleistet in Verbindung mit E 1, dass sie auf ihrem Weg von  $v_\ell$  nach  $v_{k+1}$  lediglich über der Außenfläche verläuft und keine anderen Kanten kreuzt.

Als nächstes wird die ganz rechte Kante als senkrecht fallende Kante eingefügt. Man betrachte die polygonale Kette der (direkten) Vorfahren von  $v_{k+1}$ . Die Kette war Teil von  $C_k$  und besteht daher nur aus  $T_1$ -Kanten und  $T_2$ -Kanten (Hilfssatz 5). Da eine  $T_2$ -Kante der Kette in  $\Gamma_k$  sogar eine  $T_2$ -Konturkante war, ist ihre Steigung nach E 4 kleiner als die Steigung ihrer korrespondierenden  $T_1$ -Kante. Da wiederum die Steigung der ganz linken Kante nach Konstruktionsvorschrift größer ist als die Steigung jeder anderen  $T_1$ -Kante, ist die Steigung der ganz linken Kante auch größer als die Steigung der  $T_1$ -Kanten und mittelbar der  $T_2$ -Kanten in der Kette. Damit sind die Voraussetzungen für die Anwendung von Hilfssatz 10 gegeben. Die  $T_n$ -Kanten können daher kreuzungsfrei gezeichnet werden. Auch im Fall (b.2) wird  $v_{k+1}$  durch OpInsert auf der Außenfläche ohne Kantenkreuzungen eingefügt.

Alle drei Fälle zusammen zeigen, dass OpInsert beim Einfügen von  $v_{k+1}$  keine kreuzenden Kanten verursacht.

## (2) Kreuzungsfreiheit der Kanten in OpDiagShift:

Für die Analyse von OpDiagShift werde ein Knoten  $v_t$  mit  $(v_t \neq v_{k+1})$  auf dem Konturabschnitt zwischen  $v_{k+1}$  und  $v_2$  betrachtet. Sei  $v_s$  sein Vorgänger und  $v_u$  sein Nachfolger auf dem Weg von  $v_{k+1}$  nach  $v_2$  längs der Kontur. Für  $v_2$  werde lediglich der Vorgänger betrachtet. In OpDiagShift wird  $v_t$  in Verlängerung seines  $T_1$ -Segmentes auf die Außenfläche von  $\Gamma_k$  verschoben.

(a) Als erstes werden die zu  $v_t$  inzidenten  $T_2$ -Konturkanten und die  $T_1$ -Kante  $(v_t, \text{par}_t)$  betrachtet. Für  $v_s = v_{k+1}$  ist  $(v_s, v_t)$  nach Ausführung von OpInsert eine senkrecht fallende Kante. Da für  $v_s \neq v_{k+1}$  unmittelbar vor  $v_t$  der Knoten  $v_s$  diagonal verschoben wurde, ist  $(v_s, v_t)$  auch in diesem Fall eine senkrecht fallende Kante. Die Kante  $(v_t, v_u)$  ist nach Hilfssatz 7 eine  $T_2$ -Konturkante, ihre Steigung beträgt  $\text{out}_t^2$ . Ihre korrespondierende  $T_1$ -Kante ist die Kante  $(\text{par}_t, v_t)$  und hat die Steigung  $\text{out}_t^1$ . Wegen  $v_t \neq v_{k+1}$  kann man E 4 voraussetzen und folgert  $\text{out}_t^1 > \text{out}_t^2$ . Aufgrund dieser Lage der Kanten  $(v_s, v_t)$  und  $(v_t, v_u)$  kann  $v_t$  tatsächlich so in Verlängerung von  $(\text{par}_t, v_t)$  auf die Außenfläche von  $\Gamma_k$  verschoben werden, dass weder die jetzt verlängerte Kante  $(\text{par}_t, v_t)$  noch die beiden Kanten  $(v_s, v_t)$  und  $(v_t, v_u)$  Kantenkreuzungen verursachen (Abbildung 33).

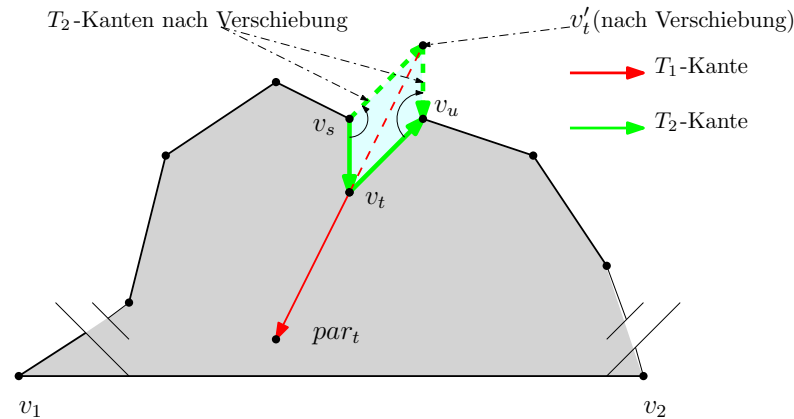
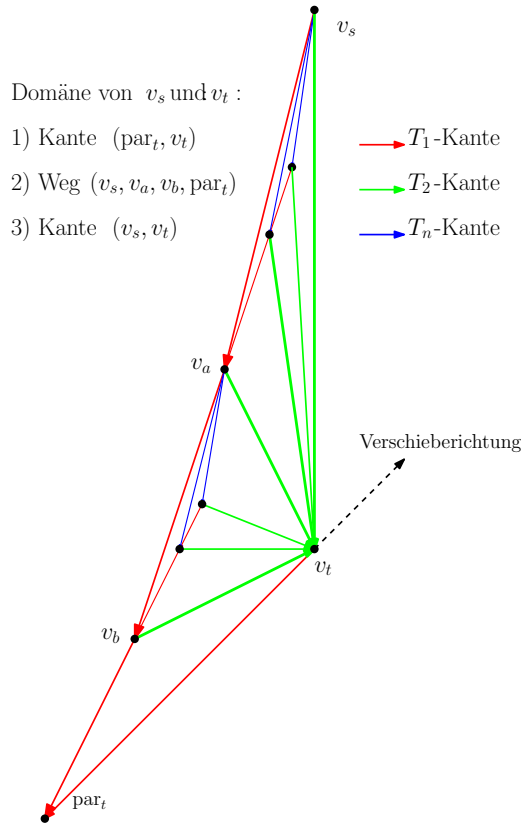


Abbildung 33: OpDiagShift im Detail, Teil 1.

(b) Betrachtet werden nun die innerhalb von  $\text{dom}_{s,t}$  auf  $v_t$  gerichteten  $T_2$ -Kanten (Abbildung 34). Aufgrund der Kantenfolge und Kantenorientierung eines Schnyder-Realisierers können sich zwischen den  $T_2$ -Kanten keine etwaigen weiteren Kanten befinden. Aus dem gleichen Grund kann zwischen der letzten dieser  $T_2$ -Kanten und der  $T_1$ -Kante  $(\text{par}_t, v_t)$  keine weitere Kante vorhanden sein. Da die Knoten entsprechend einer kanonischen Ordnung hinzugefügt wurden, ist der bisher gezeichnete Graph intern trianguliert. Die von  $v_t$  verschiedenen Endpunkte der betrachteten  $T_2$ -Kanten (und  $\text{par}_t$ ) müssen daher ihrerseits wieder durch einen Kantenzug miteinander verbunden sein. Der Kantenzug besteht aus  $T_1$ - und  $T_n$ -Kanten, er startet bei  $v_s$  und endet bei  $\text{par}_t$ .



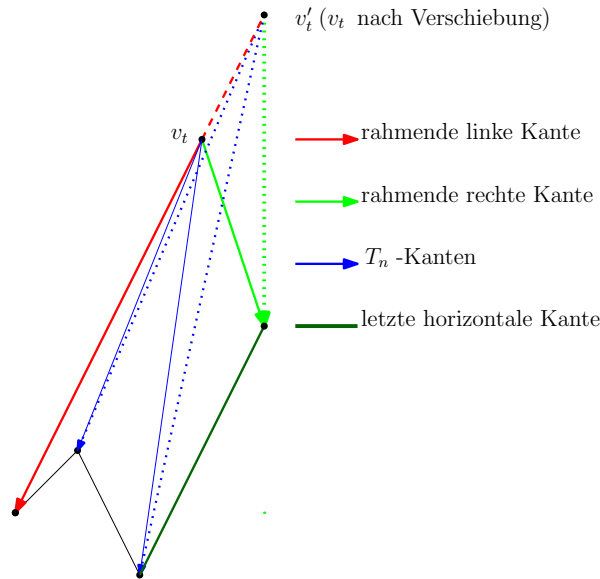
**Abbildung 34:** OpDiagShift im Detail, Teil 2.

In Abbildung 34 sind die Knoten des Kantenzuges markiert. Der Nachweis, dass die auf  $v_t$  gerichteten  $T_2$ -Kanten nach OpDiagShift keine Kreuzungen verursachen, soll mit Hilfssatz 11 erfolgen. Dazu muss nachgewiesen werden, dass sämtliche Kanten des Kantenzugs eine größere Steigung aufweisen als  $(\text{par}_t, v_t)$ . Als erstes werden unter diesem Gesichtspunkt die  $T_1$ -Kanten betrachtet. Da sich diese Kanten in der Domäne von  $v_s$  und  $v_t$  befinden, sind ihre Steigungen aufgrund E 3 größer als die von  $(\text{par}_t, v_t)$ .<sup>5</sup> Als zweites werden die  $T_n$ -Kanten des Kantenzugs beurteilt. Ihre rahmende linke Kante befindet sich ebenfalls in der Domäne von  $v_s$  und  $v_t$ . Somit ist die Steigung der rahmenden linken Kante größer als die von  $(\text{par}_t, v_t)$ . Nun wiederum sind die Steigungen der  $T_n$ -Kanten wegen E 5 größer als die

<sup>5</sup>Genaugenommen folgt diese Tatsache nur für  $v_s \neq v_{k+1}$  aus E 3. Für  $v_s = v_{k+1}$  folgt dies aus dem Beweis des Induktionsschlusses zu E 3 auf Seite 34, vgl. auch die Fußnote auf Seite 37

Steigung ihrer rahmenden linken Kante.<sup>6</sup> Daher sind die Steigungen der  $T_n$ -Kanten größer als die Steigung von  $(\text{par}_t, v_t)$ . Insgesamt sind somit die Steigungen sämtlicher Abschnitte des Kantenzugs größer als die Steigung  $(\text{par}_t, v_t)$ . Die Voraussetzungen des Hilfssatz 11 sind gegeben. Daher kann  $v_t$  in Verlängerung der Grundlinie  $(\text{par}_t, v_t)$  verschoben werden, ohne Kantenkreuzungen zu verursachen oder die Einbettung bereits eingefügter Punkte zu verändern. Damit ist der Nachweis erbracht, dass die auf  $v_t$  gerichteten  $T_2$ -Kanten nach ihrer Verschiebung in OpDiagShift keine Kreuzungen verursachen

(c) Es muss nun noch bewiesen werden, dass die  $v_t$  verlassenden  $T_n$ -Kanten nach OpDiagShift nicht die Kanten der polygonalen Kette der (direkten) Vorfahren von  $v_t$  kreuzen. Abbildung 35 zeigt das Verhalten dieser Kanten bei der Verschiebung von  $v_t$ .



**Abbildung 35:** Verschiebung der  $T_n$ -Kanten.

Hierfür soll gezeigt werden, dass nach Verschiebung von  $v_t$  die rahmende linke Kante der betrachteten  $T_n$ -Kanten (weiterhin) eine höhere Steigung aufweist als alle Kanten der polygonalen Kette der Vorfahren von  $v_t$ . Dann würde Hilfssatz 10 garantieren, dass die  $T_n$ -Kanten auch nach der Verschiebung von  $v_t$  kreuzungsfrei sind.

Das Einfügen von  $v_t$  geschah durch eine Typ-II-Einfügeoperation. Zu diesem Zeitpunkt war die Steigung der rahmenden linken Kante also größer als die aller Kanten der polygonalen Kette. Dies muss mit Ausnahme der letzten horizontalen Kante auch unmittelbar vor Verschiebung von  $v_t$  der Fall sein, weil alle Kanten der polygonalen Kette (außer der letzten horizontalen Kante) wegen E 6 nicht verändert worden sind. Es verbleibt daher der Nachweis, dass die Steigung der rahmenden linken Kante auch größer als die der letzten horizontalen Kante ist. Deren Steigung ist nach E 7 kleiner als die ihrer korrespondierenden  $T_1$ -Kante. Da die rahmende linke Kante bei ihrem Einfügen die global größte  $T_1$ -Steigung hatte, ist ihre Steigung größer als die Steigung dieser korrespondierenden Kante und damit auch

<sup>6</sup>Genaugenommen folgt auch diese Tatsache nur für  $v_s \neq v_{k+1}$  aus E 5. Für  $v_s = v_{k+1}$  folgt dies aus dem Beweis des Induktionsschlusses zu E 5 auf Seite 38, vgl. auch die Fußnote auf Seite 37

größer als die Steigung der letzten horizontalen Kante. Insgesamt ist also bewiesen, dass die Steigung der rahmenden linken Kante unmittelbar vor der Verschiebung von  $v_t$  größer ist als die Steigung jeder Kante der polygonalen Kette der Vorfahren von  $v_t$ . Die folgende Verschiebung von  $v_t$  ändert daran nichts. Dies ist darin begründet, dass die Steigung der rahmenden linken Kante bei der Verschiebung unverändert bleibt, während die Steigung der  $v_t$  verlassenden  $T_n$ -Kanten sich verringert (Abbildung 35). Hilfssatz 10 kann nun angewandt werden und garantiert, dass die  $v_t$  verlassenden  $T_n$ -Kanten auch nach der Verschiebung von  $v_t$  in OpDiagShift keine Kreuzungen mit der polygonalen Kette hervorrufen.

Weitere bisher nicht betrachtete Kanten, die zu  $v_t$  inzident sind, gibt es nicht. Damit ist der Nachweis erbracht, dass in OpDiagShift keine Kantenkreuzungen hervorgerufen werden.

### (3) Bewahrung der Einbettung:

In OpInsert wird die Einbettung bereits eingefügter Knoten offensichtlich nicht verändert. Auch in OpDiagShift ändert sich die Einbettung nicht. Anderenfalls müssten während der Verschiebung (vorübergehende) Kreuzungen oder Berührungen von Kanten auftreten. Dieses ist jedoch nicht der Fall.

Im Ergebnis wurde gezeigt, dass OpInsert und OpDiagShift weder Kreuzungen hervorrufen noch die Einbettung verändern. Daher ist die nach der  $(k + 1)$ -ten Runde vorhandene Zeichnung  $\Gamma_{k+1}$  kreuzungsfrei und bewahrt die Einbettung des entsprechenden Teilgraphen von  $G$ . Somit ist E 8 gültig in  $\Gamma_{k+1}$ .

**Eigenschaft E 9:** Entsprechend dieser Eigenschaft hat der Knoten  $v_1$  in  $\Gamma_k$  die Koordinaten  $(0,0)$ , während  $v_2$  die Koordinaten  $(k - 1, 0)$  hat. Der Knoten  $v_1$  wird in der  $(k + 1)$ -ten Runde nicht bewegt. Durch OpDiagShift wird  $v_2$  in dieser Runde um eine Längeneinheit nach rechts bewegt. Unter Verwendung der Induktionsvoraussetzung hat daher  $v_1$  die Koordinaten  $(0, 0)$  in  $\Gamma_{k+1}$ , während  $v_2$  die Koordinaten  $((k - 1) + 1, 0)$  bzw.  $(k, 0)$  in  $\Gamma_{k+1}$  hat. Dieses war im Induktionsschluss zu zeigen. Somit ist E 9 gültig in  $\Gamma_{k+1}$ .

Es wurde bewiesen, dass der Induktionsschluss im Beweis des Satzes 2 für alle Eigenschaften E 1 bis E 9 gültig ist. Er sei noch einmal wiedergegeben:

**Satz 2:** Sei  $T = (V, E)$  ein 3-Baum mit ganz linker kanonischer Ordnung  $\sigma$ .

Es bedeutet (Abbildung 22):

- (1)  $\Gamma_k$  die Zeichnung nach der  $k$ -ten Runde des Algorithmus 3TREEFEWSEG;
- (2)  $\xi_k$  die Anzahl der gezeichneten Blätter im Schnyder-Baum  $T_1$  nach der  $k$ -ten Runde;
- (3)  $s_k$  die Anzahl der Segmente, mit denen  $T_1$  nach der  $k$ -ten Runde gezeichnet ist;
- (4)  $C_k = (w_1 = v_1, \dots, w_m = v_k, \dots, w_u = v_2)$  die aktuelle Kontur;
- (5)  $C_k^* = (w_m = v_k, \dots, w_u = v_2)$  der Abschnitt der aktuellen Kontur zwischen  $v_k$  und  $v_2$ ;
- (6)  $\text{out}_i^1$  die Steigung der vom Punkt  $v_i$  ( $1 \leq i \leq k$ ) ausgehenden  $T_1$ -Kante  
( $\text{out}_i^1$  wird auch Steigung des  $T_1$ -Segments von  $v_i$  genannt);
- (7)  $\text{out}_i^2$  die Steigung der vom Punkt  $v_i$  ( $1 \leq i \leq k$ ) ausgehenden  $T_2$ -Kante;
- (8)  $\text{range}_k$  der Wertebereich der Steigungen, mit denen die  $T_1$ -Segmente gezeichnet werden.
- (9)  $\eta_k$  die größte Steigung in  $T_1$  nach der  $k$ -ten Runde.

Dann verfügt  $\Gamma_k$  für  $3 \leq k \leq n$  über die folgenden neun Eigenschaften E 1 bis E 9:

- (E 1) Die Knoten auf  $C_k$  stellen eine streng  $x$ -monotone polygonale Kette dar. Dabei sind die Knoten auf  $C_k^*$  in  $x$ -Richtung jeweils genau eine Längeneinheit voneinander entfernt.
- (E 2) Die  $T_1$ -Kanten sind mit insgesamt  $s_k = \xi_k$  unterschiedlichen  $T_1$ -Segmenten gezeichnet. Die Steigungen sind ganzzahlig mit  $\text{range}_k = (1, \eta_k)$  und  $1 \leq \eta_k \leq \xi_k$ .
- (E 3) Für einen Knoten  $v_i \neq v_2$  auf  $C_k^*$  sei  $v_j$  der Nachfolger von  $v_i$  in Richtung  $v_2$ . Dann gilt:
  - (a)  $\text{lca}_{i,j} = \text{par}_j$ .  
Das abgeschlossene Gebiet, das durch den Weg  $(v_i, \dots, \text{par}_j)$ , die Kante  $(\text{par}_j, v_j)$  und die Konturkante  $(v_i, v_j)$  begrenzt wird, ist daher wohldefiniert. Es wird Domäne der beiden Knoten  $v_i$  und  $v_j$  genannt und mit  $\text{dom}_{i,j}$  bezeichnet. Für die Domäne gilt:
  - (b) Jede  $T_1$ -Kante der Domäne hat eine Steigung, die größer oder gleich der Steigung von Kante  $(\text{par}_j, v_j)$  ist. Dabei besteht Gleichheit nur für  $(\text{par}_j, v_j)$  selbst. Für eine allgemeingültige Darstellung wird zu diesem Zweck  $\text{out}_2^1 = 0$  definiert, was mit der Verschiebung von  $v_2$  in waagerechter Richtung während OpDiagShift konsistent ist.
- (E 4) Für den Knoten  $v_i \in C_k$  einer ausgehenden  $T_2$ -Konturkante ist  $\text{out}_i^1 > \text{out}_i^2$ .
- (E 5) Die Steigung einer  $T_n$ -Kante ist größer als die Steigung ihrer rahmenden linken Kante.
- (E 6) Sei  $v_i$  ein Knoten auf  $C_k^*$ , der durch eine Typ-II-Operation eingefügt wurde. Dann gilt: Die Kanten in der polygonalen Kette der Vorfahren von  $v_i$  wurden seit seinem Einfügen mit Ausnahme der letzten horizontalen Kante nicht mehr verändert.
- (E 7) Sei  $v_i$  ein Knoten auf  $C_k^*$ , der durch eine Typ-II-Operation eingefügt wurde. Dann gilt: Die Steigung der letzten horizontalen Kante in der polygonalen Kette seiner Vorfahren ist grundsätzlich kleiner als die ihrer korrespondierenden  $T_1$ -Kante.
- (E 8) Die Zeichnung  $\Gamma_k$  ist kreuzungsfrei. Darüber hinaus wurde die Einbettung eines Knotens nach seinem Einfügen nicht mehr verändert.
- (E 9) Der Knoten  $v_1$  hat die Koordinaten  $(0,0)$ ,  $v_2$  hat die Koordinaten  $(k-1, 0)$ .

Mit Basisfall und Induktionsschluss zusammen wurde nachgewiesen, dass die Eigenschaften E 1 bis E 9 in  $\Gamma_k$  für  $3 \leq k \leq n$  gültig sind. Damit ist der Beweis von Satz 2 erbracht.  $\square$

### 2.2.5. Zentrales Ergebnis

Satz 2 kann nun dazu verwendet werden, als zentrales Ergebnis dieses Kapitels Gittergröße und visuelle Komplexität der von Algorithmus 3TREEFEWSEG angefertigten Zeichnungen zu berechnen. Das Ergebnis dieser Berechnung wird in dem folgenden Satz 3 formuliert und anschließend bewiesen. Das Kapitel schließt mit der Zeichnung eines 3-Baums als Beispiel und dem Aufzeigen einer Optimierungsmöglichkeit hinsichtlich der Gittergröße.

**Satz 3:** Algorithmus 3TREEFEWSEG fertigt von einem 3-Baum  $T = (V, E)$  mit  $|V| = n$ ,  $|E| = e$  auf dem Gitter eine Zeichnung  $\Gamma$  mit kreuzungsfreien geraden Kanten an. Die Zeichnung hat eine visuelle Komplexität von  $s = (8n - 14)/3$  bzw.  $s = 8/9 e + O(1)$ . Die Gittergröße beträgt  $O(n) \times O(n^2)$ , die Laufzeit  $O(n^2)$ .

*Beweis.* Nach Satz 2 ist  $T_1$  in  $\Gamma_n$  mit  $\xi_n$  Segmenten gezeichnet. Für den sogenannten Minimum-Realisierer beträgt die Summe aller Blätter der drei Schnyder-Bäume  $2n - 5 - \Delta_0$ , wobei  $0 \leq \Delta_0 \leq \lfloor (n - 1)/2 \rfloor$  gilt [4]. Da ein 3-Baum nach Hilfssatz 3 einen eindeutigen Schnyder-Realisierer hat, muss dieser eindeutige Realisierer auch der Minimum-Realisierer sein. In der beschriebenen Konstruktion ist  $T_1$  der Schnyder-Baum mit der geringsten Anzahl von Blättern, woraus folgt:

$$\begin{aligned}\xi_n &\leq \frac{2n - 5 - \Delta_0}{3} \\ \xi_n &\leq \frac{2n - 5}{3}.\end{aligned}$$

Die Schnyder-Bäume  $T_2$  und  $T_n$  haben jeweils  $(n - 2)$  Knoten und somit jeweils  $(n - 3)$  Kanten. Unter Berücksichtigung der drei äußeren Kanten beträgt die Anzahl der Segmente somit höchstens:

$$\begin{aligned}s_n &\leq \frac{2n - 5}{3} + n - 3 + n - 3 + 3 \\ s_n &\leq \frac{8n - 14}{3} \text{ bzw. mit } e = 3n - 6 \\ s_n &\leq 8/9 e + O(1).\end{aligned}$$

Für die visuelle Komplexität folgt daher wegen  $\Gamma = \Gamma_n$  die Behauptung.

Der Knoten  $v_2$  hat in  $\Gamma_n$  die Koordinaten  $(n - 1/0)$ . Die Zeichnung  $\Gamma$  hat daher eine Weite von  $O(n)$ . Die Kante  $(v_1, v_n)$  wird mit einer Steigung  $\text{out}_n^1$  gezeichnet, für die  $\text{out}_n^1 \leq \xi_n + 1 \leq n$  gilt. Da die  $x$ -Koordinate von  $v_n$  gleich  $(n - 2)$  ist, beträgt die Höhe von  $\Gamma$  somit  $O(n^2)$ .

Die Laufzeit in Höhe von  $O(n^2)$  setzt sich aus den folgenden Komponenten zusammen:

- (1)  $O(n)$ , um in  $n$  Runden alle Knoten einzufügen.
- (2)  $O(n)$ , um in einer Runde alle Knoten auf  $C_k^*$  diagonal zu verschieben. Die Anzahl dieser Knoten ist gleich der durchschnittlichen Weglänge in  $T_2$  von einem Knoten zur Wurzel.

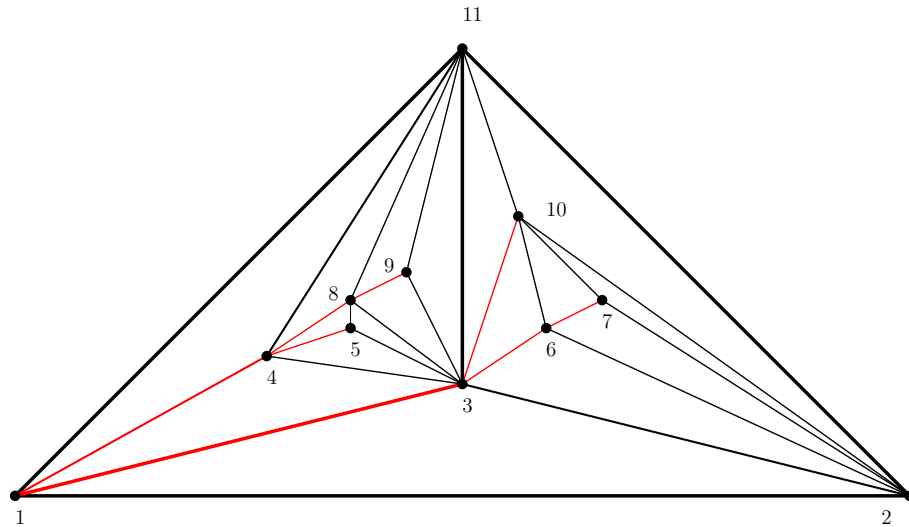
Die (triviale) Schranke hierfür ist  $O(n)$ .

Sowohl bereits berechnete Koordinaten als auch die Eingangs- und Ausgangslabes werden in entsprechenden Attributen der Knoten gespeichert. Wenn die Knoten dann zusätzlich zu ihrer Speicherung in den Schnyder-Bäumen noch in einer Datenstruktur mit konstanter Zugriffszeit wie einem Array gespeichert werden, können Berechnung und Zugriff auf Koordinaten und Steigungen insgesamt in konstanter Zeit erfolgen. Die vorbereitenden Berechnungen—wie die Konstruktion des Schnyder-Realisierers ( $O(n)$ ), der ganz linken kanonischen Ordnung ( $O(n)$ ) oder der Aufbau des zusätzlichen Arrays aus Knoten( $O(n)$ )—ändern als additive Komponenten die asymptotische Betrachtung nicht.  $\square$

**Zeichnung eines 3-Baums als Beispiel:** Der in der Abbildung 36 dargestellte 3-Baum ist so gewählt, dass der Schnyder-Baum mit der geringsten Anzahl von Blättern den Knoten 1 als Wurzel hat. Die ganz linke kanonische Ordnung lautet: 1, 2, 3, 4, 5, 8, 9, 6, 7, 10, 11. Der Schnyder-Baum  $T_1$  ist in rot gezeichnet.

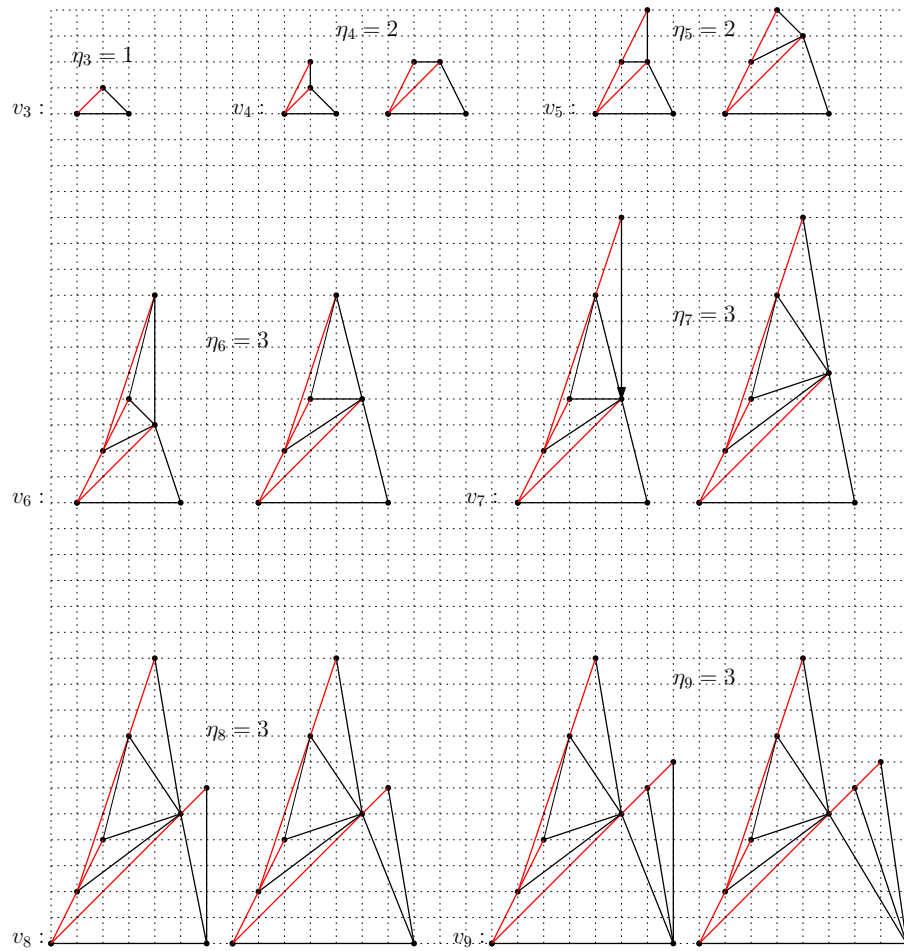
In den Abbildungen 37, 38, 39 und 40 ist die von Algorithmus 3TREEFEWSEG berechnete Zeichenfolge dargestellt. Dabei werden für jede Runde die Prozeduren OpInsert und OpDiagShift separat dargestellt. Zusätzlich ist für jede Runde der Wert des globalen Zählers  $\eta_k$  notiert.

**Optimierung:** Die Abbildungen 38 bis 40 lassen bei  $v_{10}$  Optimierungspotential hinsichtlich der Gittergröße erkennen. Die Strategie, bei einer Typ-II-Einfügeoperation das globale Maximum als Steigung zu vergeben, führt zu einem korrekten Ergebnis; sie ist jedoch noch nicht optimal. Besser ist es, bei einer Typ-II-Einfügeoperation die größte Steigung einer Kante auf der Kontur zwischen dem ganz linken Vorfahren und dem ganz rechten Vorfahren eines neu einzufügenden Knotens zu bestimmen. Dieser Wert wird dann um eins inkrementiert. Auch so ist gewährleistet, dass der neue Knoten ohne Kreuzungen eingefügt werden kann. Die asymptotische Gitterschranke von  $O(n) \times O(n^2)$  wird jedoch durch diese Optimierung nicht verbessert.

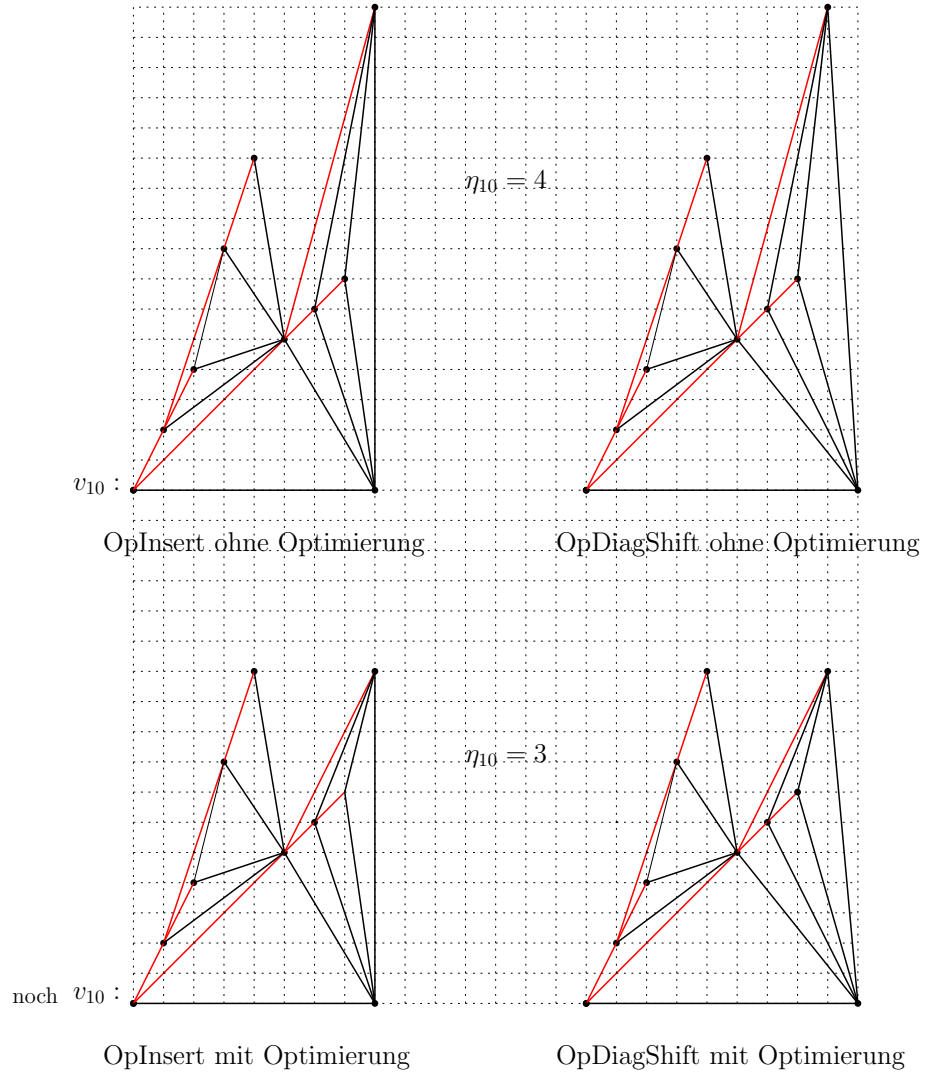


**Abbildung 36:** Der zu zeichnende 3-Baum.

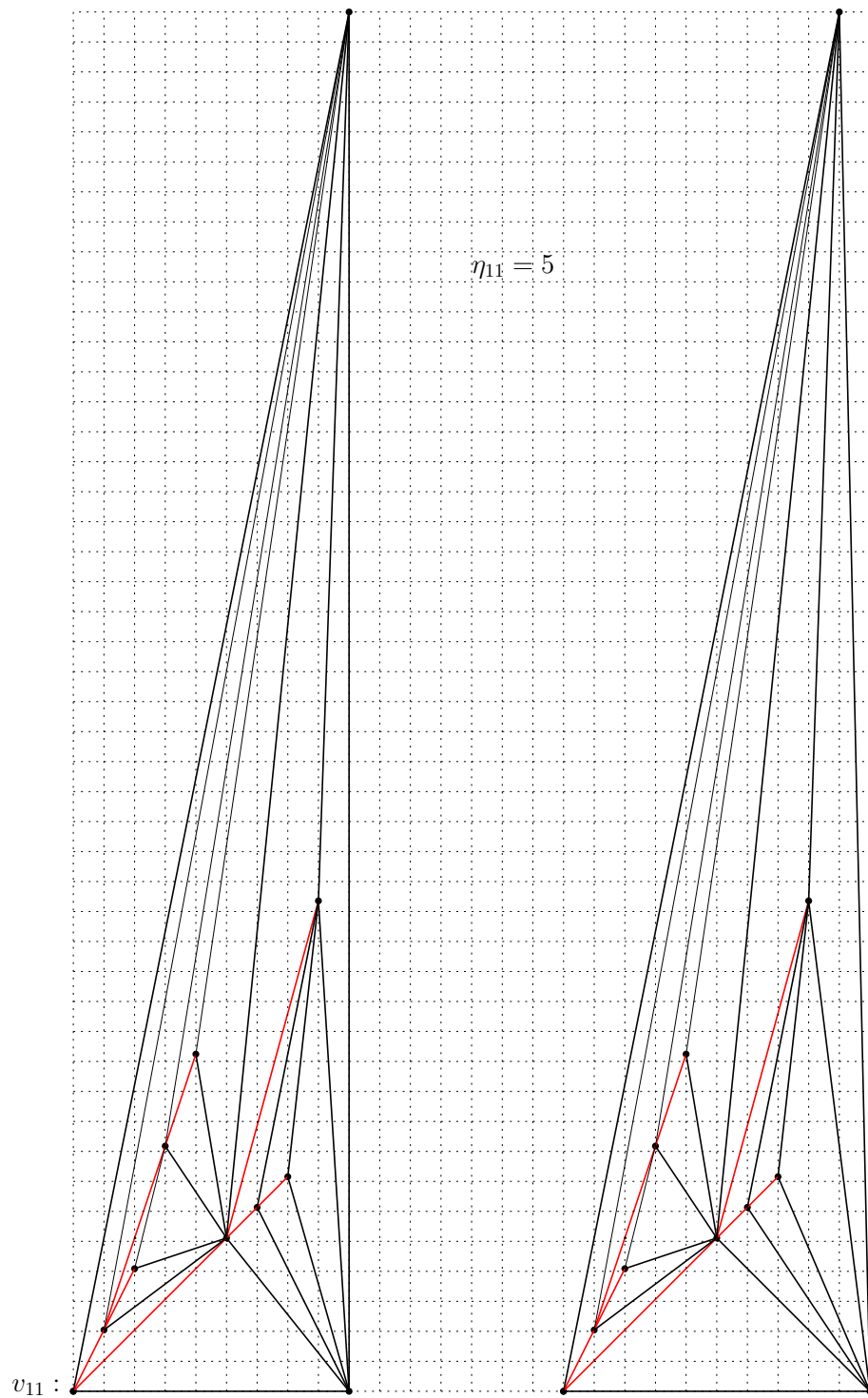




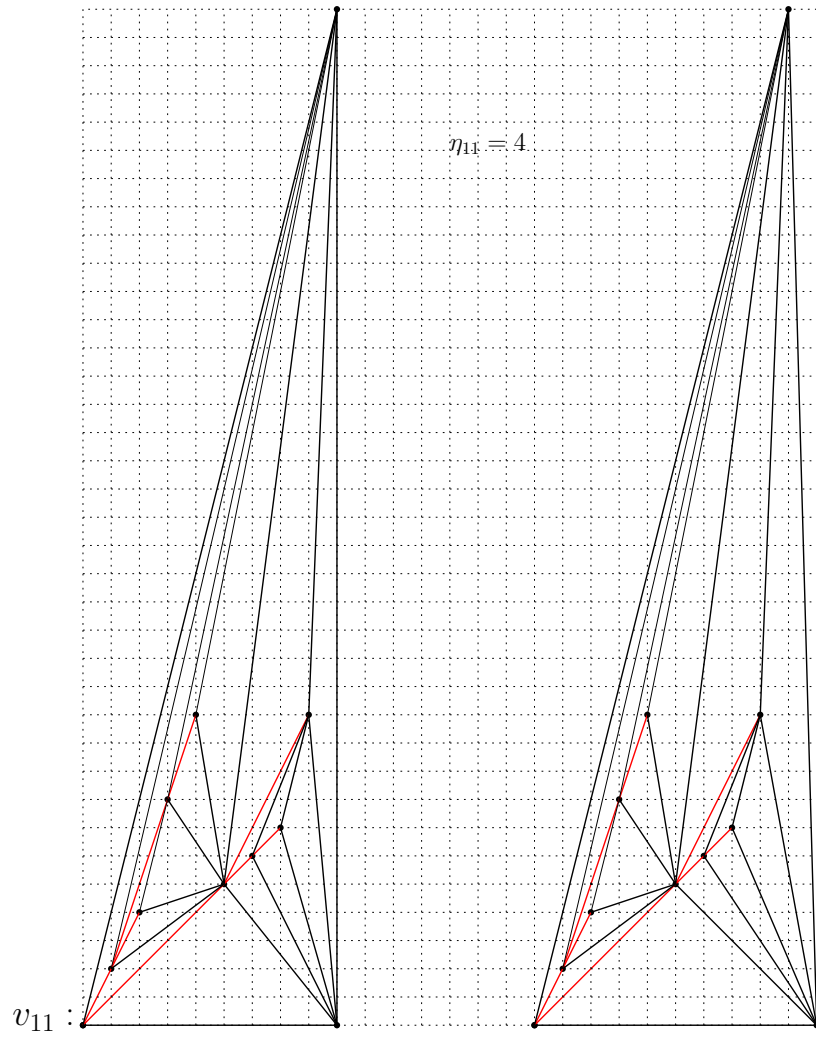
**Abbildung 37:** Zeichnungen  $\Gamma_3$  bis  $\Gamma_9$ .



**Abbildung 38:** Zeichnung  $\Gamma_{10}$ , zusätzlich mit möglicher Optimierung.



**Abbildung 39:** Zeichnung  $\Gamma_{11}$  ohne vorangegangene Optimierung.



**Abbildung 40:** Zeichnung  $\Gamma_{11}$  mit vorangegangener Optimierung.

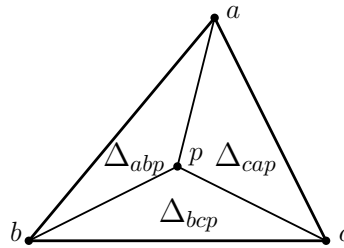
## 2.3. Segmentminimale Gitterzeichnungen von vollständig balancierten 3-Bäumen

Den rekursiven Aufbau eines 3-Baums kann man durch seinen sogenannten repräsentativen Baum eindeutig beschreiben [33]. Es wird zunächst ein Algorithmus beschrieben, der von einem 3-Baum eine segmentminimale Gitterzeichnung erstellt, deren Fläche exponentiell zur Höhe des repräsentativen Baums ist. Bei der induktiven Erzeugung eines 3-Baums entsteht ein vollständig balancierter 3-Baum, wenn in allen Rekursionsschritten auf allen inneren Dreiecksflächen eine Stapeloperation ausgeführt wird. Wenn man die Eingabe für den zuvor erwähnten Algorithmus auf vollständig balancierte 3-Bäume beschränkt, fertigt dieser Algorithmus eine segmentminimale Zeichnung des vollständig balancierten 3-Baums auf einem Gitter der Größe  $O(n) \times O(n)$  an. Für Beschreibung und Analyse des Algorithmus werden die Definition des repräsentativen Baums und einige Eigenschaften von baryzentrischen Koordinaten benötigt.

### 2.3.1. Spezifische Grundlagen

**Definition 11 ([33]):** Der repräsentative Knoten eines 3-Baums  $T = (V, E)$  mit  $|V| > 3$  ist der eindeutige gemeinsame Nachbar der drei Knoten auf der Außenfläche des 3-Baums (Abbildung 41).

**Definition 12 ([33]):** Sei  $p$  der repräsentative Knoten eines 3-Baums  $T$  und seien  $a, b, c$  die drei Knoten auf der Außenfläche von  $T$ . Der Knoten  $p$  und die drei äußeren Knoten  $a, b, c$  bilden zusammen die drei Dreiecke  $\Delta_{abp}, \Delta_{bcp}, \Delta_{cap}$  (Abbildung 41). Diese drei Dreiecke nennt man die um  $p$  herum verschachtelten Dreiecke. Aus dem vorangegangenen Kapitel ist bekannt, dass für einen 3-Baum  $T$  der durch ein beliebiges Dreieck  $\Delta$  induzierte Untergraph  $T(\Delta)$  von  $T$  ebenfalls wieder ein 3-Baum ist (Hilfssatz 8(c)). Dies gilt insbesondere auch für die Untergraphen  $T(\Delta_{abp}), T(\Delta_{bcp})$  und  $T(\Delta_{cap})$ .

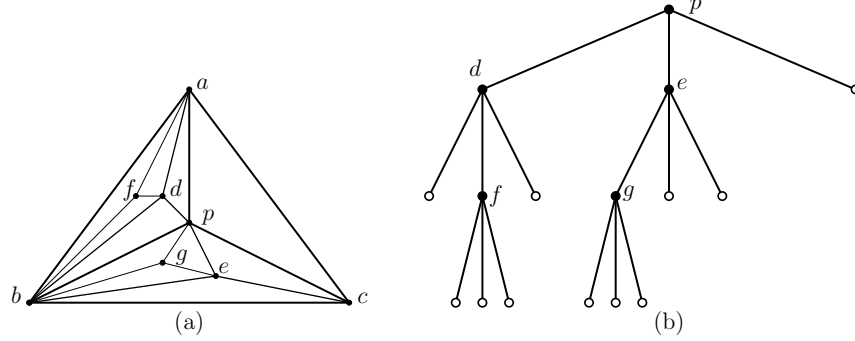


**Abbildung 41:** Repräsentativer Knoten  $p$  und die um  $p$  herum verschachtelten Dreiecke, nach [33].

**Definition 13 ([33]):** Der repräsentative Baum  $T_R$  eines 3-Baums  $T = (V, E)$  mit  $|V| = n$  wird induktiv wie folgt definiert:

- (a) Für  $n = 3$  besteht  $T_R$  aus einem einzigen Knoten.
- (b) Für  $n > 3$  ist die Wurzel  $p$  von  $T_R$  gleich dem repräsentativen Knoten von  $T$ . Seien  $q_1, q_2, q_3$  die drei gegen den Uhrzeigersinn angeordneten Kindknoten von  $p$  in  $T_R$ . Die an

diesen drei Knoten verwurzelten Unterbäume von  $T_R$  sind die repräsentativen Bäume von  $T(\Delta_1)$ ,  $T(\Delta_2)$  und  $T(\Delta_3)$ , wobei  $\Delta_1$ ,  $\Delta_2$  und  $\Delta_3$  die drei um  $p$  herum verschachtelten Dreiecke sind. Dabei seien die drei Dreiecke ebenfalls gegen den Uhrzeigersinn angeordnet (Abbildung 42).



**Abbildung 42:** (a) Ein 3-Baum und (b) sein repräsentativer Baum  $T_R$ , in Anlehnung an [33].

**Baryzentrische Koordinaten:** Mit baryzentrischen Koordinaten können die Koordinaten eines Punktes relativ zu einem gegebenen Dreieck elegant angegeben werden. Allgemein bekannte Eigenschaften sind (z.B. [9]):

- (1) Die baryzentrischen Koordinaten  $(\alpha/\beta/\gamma)$  eines Punktes  $P$  seien mit Bezug auf das Dreieck  $\Delta_{ABC}$  gegeben. Unter Verwendung von Ortsvektoren erhält man die kartesischen Koordinaten von  $P$  mit Hilfe der Gleichung

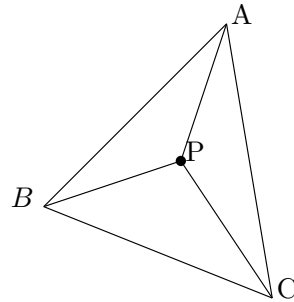
$$\vec{p} = \alpha \vec{a} + \beta \vec{b} + \gamma \vec{c}.$$

- (2) Der Schwerpunkt eines Dreiecks weist die baryzentrischen Koordinaten  $(\frac{1}{3}/\frac{1}{3}/\frac{1}{3})$  auf.
- (3) Die baryzentrischen Koordinaten  $(\alpha/\beta/\gamma)$  eines Punktes  $P$  lassen sich als Quotienten von Dreiecksflächen darstellen. Für die Analyse des Algorithmus ist lediglich der Fall relevant, dass  $P$  innerhalb des Referenzdreiecks  $\Delta_{ABC}$  liegt. Dann lässt sich auf eine Orientierung der Dreiecksflächen verzichten und es gilt (Abb. 43):

$$\alpha = \frac{\text{Fläche}(\Delta_{BCP})}{\text{Fläche}(\Delta_{ABC})}$$

$$\beta = \frac{\text{Fläche}(\Delta_{CAP})}{\text{Fläche}(\Delta_{ABC})}$$

$$\gamma = \frac{\text{Fläche}(\Delta_{ABP})}{\text{Fläche}(\Delta_{ABC})}$$



**Abbildung 43:** Baryzentrische Koordinaten.

- (4) Drei Punkte  $P_i$ ,  $i = 1, 2, 3$  mit den baryzentrischen Koordinaten  $(\alpha_i/\beta_i/\gamma_i)$  sind genau dann kollinear, wenn

$$\begin{vmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{vmatrix} = 0 \text{ ist.}$$

### 2.3.2. Algorithmus 3TREEOPTSEG

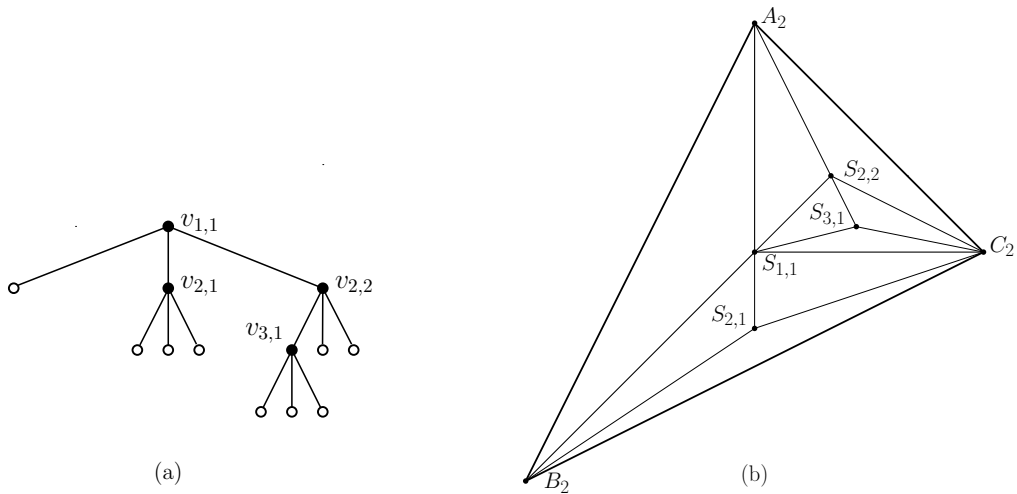
Es wird zunächst Algorithmus 3TREEOPTSEG beschrieben, der von einem beliebigen 3-Baum eine segmentoptimale Gitterzeichnung erstellt, deren benötigte Fläche exponentiell zur Höhe des korrespondierenden repräsentativen Baums ist. Im Einzelnen gilt für diesen Algorithmus:

**Eingabe:** Ein 3-Baum  $T$  mit  $n$  Knoten ( $n > 3$ ) und  $e$  Kanten, dessen Struktur durch seinen repräsentativen Baum  $T_R$  gegeben ist. Die Höhe von  $T_R$  sei mit  $h$  bezeichnet. Ferner sei  $i$  die  $i$ -te Hierarchieebene von  $T_R$ . Die Anzahl der Knoten auf der  $i$ -ten Hierarchieebene werde mit  $\xi_i$  bezeichnet; die Knoten der  $i$ -ten Hierarchieebene seien mit  $v_{i,j}$  fortlaufend nummeriert, wobei  $1 \leq j \leq \xi_i$  gelte.

**Ausgabe:** Eine auf dem Gitter mit kreuzungsfreien geraden Kanten realisierte Zeichnung  $\Gamma$  von  $T$  mit der optimalen Anzahl von Segmenten  $s = 2n - 2$ . Die Größe des Gitters beträgt  $2 \cdot 3^{h-1} \times 2 \cdot 3^{h-1}$ .

**Beschreibung:** Der Algorithmus verwendet modifiziert eine in [46] beschriebene rekursiv erzeugte Folge  $\Gamma_i$  von 3-Baum-Zeichnungen:

- (1)  $\Gamma_1$  ist die Zeichnung des  $K_4$  als Dreieck mit den Eckpunkten  $A(0/1)$ ,  $B(-1/-1)$ ,  $C(1/0)$  und dem mit  $S_{1,1}$  bezeichneten Schwerpunkt  $(0/0)$  dieses Dreiecks.
- (2) Für  $2 \leq i \leq h$  geht  $\Gamma_i$  aus  $\Gamma_{i-1}$  durch folgende zwei Konstruktionsschritte hervor:
  - (a) Multiplizieren aller Koordinaten mit 3.
  - (b) Für jeden Knoten  $v_{i,j}$  ( $1 \leq j \leq \xi_i$ ) der  $i$ -ten Hierarchieebene von  $T_R$  wird ein Punkt  $S_{i,j}$  in den Schwerpunkt des zu  $v_{i,j}$  korrespondierenden Dreiecks (entsprechend des repräsentativen Baums) positioniert und mit den Eckpunkten dieses Dreiecks verbunden (Abbildungen 44(a) und (b)).



**Abbildung 44:** (a) Repräsentativer Baum. (b) Die zugehörige Zeichnung  $\Gamma_3$ .

Die so definierte Folge der Zeichnungen  $\Gamma_i$  verfügt über die beiden nachstehenden Eigenschaften (E1) und (E2):

**(E 1)** Die Knoten aller Zeichnungen  $\Gamma_i, 1 \leq i \leq h$  weisen ganzzahlige Koordinaten auf. Die Zeichnungen  $\Gamma_i$  sind somit Gitterzeichnungen. Der Beweis wird über vollständige Induktion geführt:

*Beweis.* Der Nachweis des Basisfalls erfolgt unmittelbar aus der Definition von  $\Gamma_1$ .

Induktionsschluss: Sei die Behauptung für  $i - 1$  bewiesen. Nach Induktionsvoraussetzung haben dann  $A, B, C$  sowie die  $S_{p,q}$  mit  $1 \leq p \leq i - 1$  und  $q \leq \xi_p$  in  $\Gamma_{i-1}$  ganzzahlige Koordinaten. Da ihre Koordinaten in  $\Gamma_i$  aus ihren Koordinaten in  $\Gamma_{i-1}$  durch Multiplikation mit drei hervorgehen, sind ihre Koordinaten auch in  $\Gamma_i$  ganzzahlig. Verbleiben in  $\Gamma_i$  die Punkte  $S_{i,j}$  mit  $j \leq \xi_i$ . Ohne Beschränkung der Allgemeinheit werde der weitere Beweis für einen dieser Punkte geführt, er sei mit  $S$  bezeichnet. Aus der Konstruktionsvorschrift folgt, dass  $S$  Schwerpunkt eines in  $\Gamma_{i-1}$  bestehenden Dreiecks ist. Seien  $P, Q, R$  durch Umbenennung die Eckpunkte dieses Dreiecks; ferner seien  $\vec{p}_{i-1}, \vec{q}_{i-1}, \vec{r}_{i-1}$  die Ortsvektoren dieser Punkte in  $\Gamma_{i-1}$  und  $\vec{p}_i, \vec{q}_i, \vec{r}_i$  die Ortsvektoren in  $\Gamma_i$ . Dieses Dreieck sei nun das Referenzdreieck für baryzentrische Koordinaten; für die kartesischen Koordinaten von  $S$  in  $\Gamma_i$  gilt daher

$$\begin{aligned}\vec{s} &= 1/3\vec{p}_i + 1/3\vec{q}_i + 1/3\vec{r}_i \\ \vec{s} &= 1/3 \cdot 3\vec{p}_{i-1} + 1/3 \cdot 3\vec{q}_{i-1} + 1/3 \cdot 3\vec{r}_{i-1}\end{aligned}$$

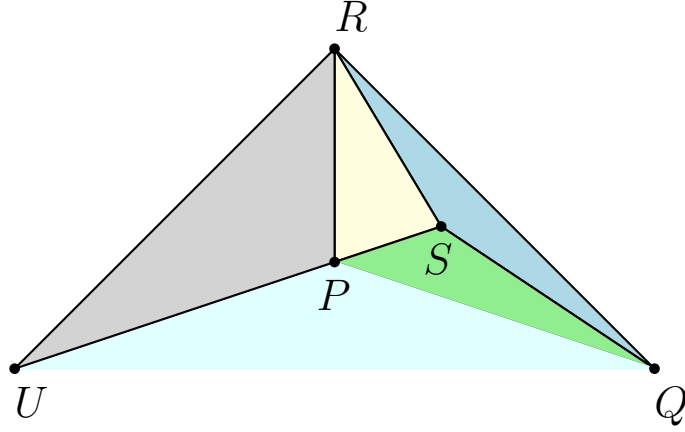
Da nach Induktionsvoraussetzung  $\vec{p}_{i-1}, \vec{q}_{i-1}$  und  $\vec{r}_{i-1}$  ganzzahlige Koordinaten haben, haben auch  $\vec{s}$  bzw.  $S$  ganzzahlige Koordinaten. Der Induktionsschluss ist somit bewiesen.

Mit Basisfall und Induktionsschluss ist auch die Behauptung bewiesen. Alle  $\Gamma_i, 1 \leq i \leq h$  sind daher Gitterzeichnungen.  $\square$

**(E 2)** Während der Konstruktion von  $\Gamma_i, i \geq 2$  werden bei der Positionierung eines Knoten  $S$  in den Schwerpunkt eines Dreiecks  $\Delta$  zwei zusätzliche Segmente erzeugt.

*Beweis.* Wie zuvor seien durch Umbenennung die Eckpunkte von  $\Delta$  mit  $P, Q, R$  bezeichnet. Das Dreieck  $\Delta$  selbst ist ebenfalls durch eine vorangegangene Stapeloperation entstanden. Ohne Beschränkung der Allgemeinheit sei  $P$  derjenige der drei Eckpunkte von  $\Delta$ , der in dieser Stapeloperation der Zeichnung hinzugefügt wurde. Ferner sei mit  $\Delta'$  das Dreieck bezeichnet, auf dessen Schwerpunkt  $P$  während dieser Stapeloperation positioniert wurde. Zwei der drei Eckpunkte von  $\Delta'$  sind  $Q$  und  $R$ . Der dritte Eckpunkt von  $\Delta'$  sei mit  $U$  bezeichnet. Der Punkt  $P$  befindet sich also im Schwerpunkt des Dreiecks  $\Delta'$  mit den Eckpunkten  $U, Q, R$ ; der Knoten  $S$  hingegen befindet sich im Schwerpunkt des Dreiecks  $\Delta$  mit den Eckpunkten  $P, Q, R$ . Insgesamt ergibt sich also eine Anordnung der Knoten, wie sie in (Abbildung 45) dargestellt ist. Das Dreieck  $\Delta_{UQR}$  sei nun das Referenzdreieck für baryzentrische Koordinaten. Dann folgt:





**Abbildung 45:** Kollinearität von  $U, P, S$ .

- (a)  $U$  hat die baryzentrischen Koordinaten  $(1/0/0)$ .
- (b)  $P$  hat die baryzentrischen Koordinaten  $(\frac{1}{3}/\frac{1}{3}/\frac{1}{3})$ .
- (c)  $S$  hat die baryzentrischen Koordinaten

$$\alpha_s = \frac{\text{Fläche}(\Delta_{SQR})}{\text{Fläche}(\Delta_{UQR})} = \frac{1/3 \cdot \text{Fläche}(\Delta_{PQR})}{\text{Fläche}(\Delta_{UQR})} = \frac{1/3 \cdot 1/3 \cdot \text{Fläche}(\Delta_{UQR})}{\text{Fläche}(\Delta_{UQR})} = 1/9,$$

$$\begin{aligned} \beta_s &= \frac{\text{Fläche}(\Delta_{UPR}) + \text{Fläche}(\Delta_{PSR})}{\text{Fläche}(\Delta_{UQR})} = \frac{1/3 \cdot \text{Fläche}(\Delta_{UQR}) + 1/3 \cdot \text{Fläche}(\Delta_{PQR})}{\text{Fläche}(\Delta_{UQR})} \\ &= \frac{1/3 \cdot \text{Fläche}(\Delta_{UQR}) + 1/9 \cdot \text{Fläche}(\Delta_{UQR})}{\text{Fläche}(\Delta_{UQR})} = 4/9, \end{aligned}$$

$$\begin{aligned} \gamma_s &= \frac{\text{Fläche}(\Delta_{UQP}) + \text{Fläche}(\Delta_{PQS})}{\text{Fläche}(\Delta_{UQR})} = \frac{1/3 \cdot \text{Fläche}(\Delta_{UQR}) + 1/3 \cdot \text{Fläche}(\Delta_{PQR})}{\text{Fläche}(\Delta_{UQR})} \\ &= \frac{1/3 \cdot \text{Fläche}(\Delta_{UQR}) + 1/9 \cdot \text{Fläche}(\Delta_{UQR})}{\text{Fläche}(\Delta_{UQR})} = 4/9. \end{aligned}$$

Da die Determinante

$$\begin{vmatrix} 1 & 0 & 0 \\ 1/3 & 1/3 & 1/3 \\ 1/9 & 4/9 & 4/9 \end{vmatrix} = 0 \text{ ist,}$$

sind die Punkte  $U, P, S$  kollinear. Das Positionieren eines Knoten erzeugt also zwei zusätzliche Segmente, was zu beweisen war.  $\square$

Mit den beiden Eigenschaften (E1) und (E2) werden nun visuelle Komplexität und Gitterschranke der Zeichnung bestimmt, die durch Algorithmus 3TREEOPTSEG von  $T$  erstellt wird.

**Visuelle Komplexität:** Mit der Eigenschaft E 2 kann nun bewiesen werden, dass die von Algorithmus 3TREEOPTSEG erzielte Segmentanzahl  $s = 2n - 2$  beträgt. Der Beweis erfolgt wiederum mit vollständiger Induktion.

*Beweis.* Der Basisfall besteht hier für  $n = 4$ . Da der  $K_4$  sechs Kanten hat, ist die Aussage für den Basisfall wahr.

Induktionsschluss: Sei die Aussage für  $n$  Knoten wahr. Beim Positionieren des  $(n + 1)$ -ten Knoten werden wie zuvor bewiesen zwei zusätzliche Segmente erzeugt. Daher beträgt die Segmentanzahl für  $(n + 1)$  Knoten

$$s = (2n - 2) + 2 = 2(n + 1) - 2.$$

Damit ist der Induktionsschluss bewiesen. Mit Basisfall und Induktionsschluss ist auch der Beweis für die Behauptung erbracht.  $\square$

Nach [12] ist die bewiesene Segmentanzahl von  $s = 2n - 2$  optimal. Algorithmus 3TREEOPTSEG erstellt also segmentminimale Zeichnungen von dem als Eingabe dienenden 3-Baum  $T$ .

**Gittergröße:** Die Struktur von  $T$  wird durch seinen repräsentativen Baum beschrieben, dessen Höhe mit  $h$  bezeichnet wurde. Algorithmus 3TREEOPTSEG fertigt für  $T$  entsprechend dem repräsentativen Baum die zuvor beschriebene Folge  $\Gamma_i$  von Zeichnungen an. Die Zeichnung  $\Gamma_h$  stellt dann offensichtlich eine Gitterzeichnung  $\Gamma$  von  $T$  dar. Man ordne nun einem repräsentativen Baum, der aus nur einem einzigen Knoten besteht, die Höhe eins zu. Die Gitterschranke  $2 \cdot 3^{h-1} \times 2 \cdot 3^{h-1}$  folgt dann sofort aus der Konstruktionsvorschrift für die  $\Gamma_i$ .

### 2.3.3. Algorithmus BALANCED3TREEOPTSEG

Seien nun als Eingabe für Algorithmus 3TREEOPTSEG nur vollständig balancierte 3-Bäume zugelassen. Für diesen Fall wird der Algorithmus in Algorithmus BALANCED3TREEOPTSEG umbenannt. Die Höhe des repräsentativen Baums beträgt dann  $h = O(\log n)$ . Aus der zuvor für Algorithmus 3TREEOPTSEG berechneten Gitterschranke von  $2 \cdot 3^{h-1} \times 2 \cdot 3^{h-1}$  ergibt sich für Algorithmus BALANCED3TREEOPTSEG unmittelbar eine Gitterschranke von  $O(n) \times O(n)$ . Das zentrale Ergebnis dieses Kapitels lässt sich somit im folgenden Satz 4 zusammenfassen:

**Satz 4:** Algorithmus BALANCED3TREEOPTSEG fertigt für vollständig balancierte 3-Bäume mit  $n$  Knoten ( $n > 3$ ) eine segmentoptimale Zeichnung auf einem  $O(n) \times O(n)$  großen Gitter an.

## 2.4. Segmentreduzierte Gitterzeichnungen von maximalen außenplanaren Graphen

Ein zusammenhängender Graph mit mindestens drei Knoten heißt außenplanar, wenn er so in die Ebene eingebettet werden kann, dass alle seine Knoten auf der Außenfläche liegen [7]. Damit weisen außenplanare Graphen eine die Struktur vereinfachende Eigenschaft auf. Dies zeigt sich auch darin, dass es Algorithmen gibt, die außenplanare Graphen auf einem Gitter mit subquadratischer Gitterschranke zeichnen [10][18]. Ein außenplanarer Graph  $G$  ist maximal, wenn  $G$  nach Hinzufügen einer weiteren Kante nicht mehr außenplanar ist. In diesem Kapitel wird für maximale außenplanare Graphen zunächst eine Dekompositionstechnik und anschließend ein Algorithmus für die Anfertigung segmentreduzierter Gitterzeichnungen beschrieben.

### 2.4.1. Spezifische Grundlagen

**Definition 14 ([30]):** Ein maximaler außenplanarer Graph kann rekursiv wie folgt definiert werden:

- (1) Der Graph  $K_3$ , bestehend aus drei wechselseitig miteinander verbundenen Knoten, ist ein maximaler außenplanarer Graph.
- (2) Sei  $G$  ein maximaler außenplanarer Graph, der so in die Ebene eingebettet ist, dass jeder Knoten auf der Außenfläche liegt. Sei  $H$  ein Graph, den man erhält, indem man einen neuen Knoten auf der Außenfläche von  $G$  einfügt und ihn mit den zwei Knoten einer ebenfalls auf der Außenfläche gelegenen Kante von  $G$  verbindet. Dann ist  $H$  ebenfalls maximal außenplanar.
- (3) Ein Graph ist genau dann ein maximaler außenplanarer Graph, wenn er durch eine endliche Anzahl von Anwendungen der Aussagen (1) und (2) entstanden ist.

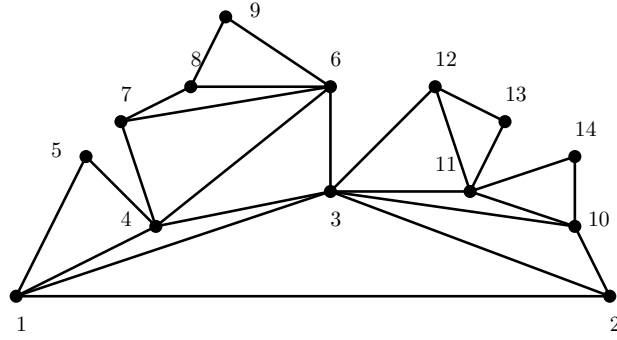
Diese rekursive Definition führt zu einer kanonischen Ordnung der Knoten.

**Definition 15 (nach [30]):** Sei  $G = (V, E)$  mit  $|V| = n$  ein maximaler außenplanarer Graph. Eine Ordnung  $\sigma = (v_1, v_2, \dots, v_n)$  der Knoten von  $G$  ist eine kanonische Ordnung, wenn die beiden folgenden Bedingungen erfüllt sind:

- (1) Die Knoten  $v_1, v_2$  und  $v_3$  formen ein Dreieck in  $G$ .
- (2) Sei  $N$  die Menge der Nachbarknoten eines Knotens  $v_k$  mit  $3 \leq k \leq n$ . Dann existieren genau zwei Nachbarknoten  $v_i, v_j \in N$ , für die  $i < k$  und  $j < k$  gilt.

Die kanonische Ordnung eines maximalen außenplanaren Graphen lässt die Reihenfolge erkennbar werden, in der die Knoten während der Konstruktion in den Graphen eingefügt werden. Von Interesse sind insbesondere die Ordnungen, in denen die Kante  $(v_1, v_2)$  auf der Außenfläche liegt. Abbildung 46 zeigt eine solche kanonische Ordnung.

**Definition 16 ([3]):** Sind zwei Knoten weder Vorfahre noch Nachfahre des jeweils anderen, so sind sie nicht miteinander verwandt.



**Abbildung 46:** Kanonische Ordnung für einen maximal außenplanaren Graphen.

**Hilfssatz 13 ([30]):** Für einen maximalen außenplanaren Graph  $G = (V, E)$  gilt:

- (a) Sei 2-Knoten die Bezeichnung für einen Knoten, der Grad zwei hat und dessen beide Nachbarn durch eine Kante miteinander verbunden sind. Dann hat  $G$  mindestens zwei 2-Knoten.
- (b) Der Graph  $G'$ , der aus  $G$  durch Entfernen eines 2-Knotens hervorgeht, ist selbst wieder maximal außenplanar.

**Hilfssatz 14:** Sei  $G = (V, E)$  mit  $|V| = n$  ein maximaler außenplanarer Graph mit zwei beliebig gewählten benachbarten Knoten  $v_1$  und  $v_2$ . Dann enthält  $G$  (mindestens) einen 2-Knoten, der von  $v_1$  und  $v_2$  verschieden ist.

*Beweis.* Der Beweis für die Existenz eines solchen 2-Knotens wird induktiv über die Anzahl der Knoten geführt.

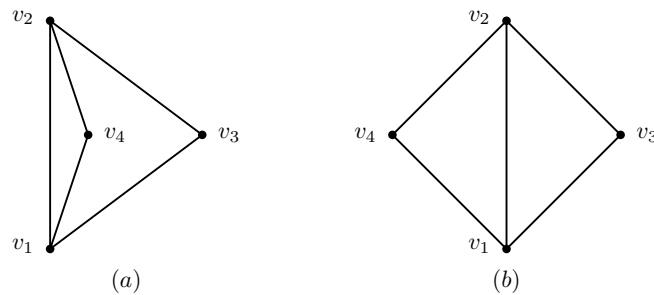
- (1) Basisfall: Für  $n = 3$  ist die Aussage offensichtlich wahr.
- (2) Induktionsschluss: Sei die Aussage für einen maximalen außenplanaren Graphen  $G_n$  mit  $n$  Knoten bewiesen. Ein maximaler außenplanarer Graph  $G_{n+1}$  mit  $(n + 1)$  Knoten geht aus  $G_n$  durch die in Definition 14 beschriebene Konstruktion hervor. Man wähle nun zwei (beliebige) benachbarte Knoten  $v_1$  und  $v_2$ . Nach Induktionsvoraussetzung weist  $G_n$  einen 2-Knoten  $v_i$  auf, der von  $v_1$  und  $v_2$  verschieden ist. Es werden nun zwei Fälle unterschieden.
  - (a) Die Knoten  $v_{n+1}$  und  $v_i$  sind nicht benachbart. Dann ist  $v_i$  auch in  $G_{n+1}$  ein 2-Knoten und nichts weiter zu beweisen.
  - (b) Die Knoten  $v_{n+1}$  und  $v_i$  sind benachbart. Dann ist  $v_i$  in  $G_{n+1}$  kein 2-Knoten mehr. Man unterscheidet nun weiter:
    - (i) Der Knoten  $v_{n+1}$  ist ungleich  $v_1$  und  $v_2$ .  
Da nach Definition 14  $v_{n+1}$  den Grad zwei hat und seine beiden Nachbarn durch eine Kante verbunden sind, ist  $v_{n+1}$  ein 2-Knoten. Vereinbarungsgemäß ist er von  $v_1$  und  $v_2$  verschieden. Der Induktionsschluss ist daher für diesen Fall bewiesen.
    - (ii) Der Knoten  $v_{n+1}$  ist gleich  $v_1$  oder  $v_2$ .  
Zwei 2-Knoten können nicht durch eine Kante verbunden sein. Sonst wäre diese Kante bei der Positionierung des zweiten Knotens hinzugefügt worden und der zuerst eingefügte Knoten hätte den Grad drei. Es sei nun ohne Beschränkung der

Allgemeinheit  $v_{n+1}$  gleich  $v_1$ . Wie zuvor beschrieben ist  $v_{n+1}$  ein 2-Knoten. Nach Hilfssatz 13(a) existiert in  $G_{n+1}$  ein weiterer 2-Knoten. Aufgrund der einleitenden Bemerkung kann dieser 2-Knoten nicht mit  $v_1 = v_{n+1}$  durch eine Kante verbunden sein. Er kann daher nicht  $v_2$  sein. Somit ist dieser Knoten ein 2-Knoten und von  $v_1$  und  $v_2$  verschieden, was zu beweisen war. Auch für diesen letzten Fall wurde daher der Induktionsschluss bewiesen.

Alle Fälle zusammen ergeben die Gültigkeit des Induktionsschlusses.

(3) Mit Basisfall und Induktionsschluss ist die Behauptung bewiesen.  $\square$

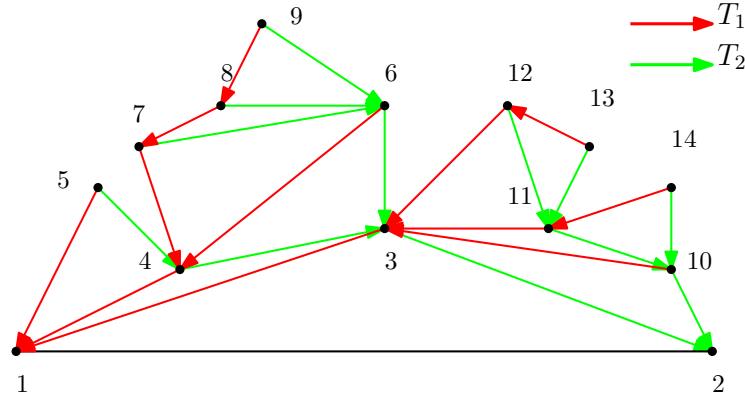
Es gibt maximale außenplanare Graphen, die so eingebettet werden können, dass nicht alle Knoten auf der Außenfläche liegen. Ein einfaches Beispiel zeigt Abbildung 47. Die weiteren Ausführungen setzen jedoch voraus, dass alle Knoten auf dem Rand der Außenfläche liegen. Ist dies nicht der Fall, kann eine solche Einbettung vorab bestimmt werden. Die Reihenfolge der Knoten auf dem Rand der Außenfläche entspricht hierbei dem Hamiltonkreis des Graphen und kann in linearer Zeit berechnet werden [31].



**Abbildung 47:** Zwei Einbettungen eines maximalen außenplanaren Graphen. (a) Nicht alle Knoten liegen auf der Außenfläche. (b) Alle Knoten liegen auf der Außenfläche.

**Hilfssatz 15 (vgl. [17]):** Sei  $G = (V, E)$  mit  $|V| = n$  ein maximaler außenplanarer Graph mit kanonischer Ordnung  $\sigma$  und  $(v_1, v_2)$  eine Kante des Graphen auf der Außenfläche. Durch  $\sigma$  wird eine Einteilung von  $E \setminus \{(v_1, v_2)\}$  in zwei Bäume induziert, so dass jeder Knoten  $v \in V \setminus \{v_1, v_2\}$  in beiden Bäumen enthalten ist und die Kanten in Richtung zur Wurzel orientiert sind. Die mit  $T_1$  bzw.  $T_2$  bezeichneten Bäume sind in  $v_1$  bzw.  $v_2$  verwurzelt. Abbildung 48 zeigt für den bereits bekannten Graphen die Bäume  $T_1$  und  $T_2$ . Dieser Satz ist bereits aus [17] bekannt. Die dortige Beweisidee deutet einen Schnyder-Realisierer als ein Mengensystem, dessen Elemente partiell geordnete Mengen von Kanten des betrachteten Graphen sind. Der folgende Beweis dagegen ist eher konstruktiv orientiert.

*Beweis.* Implizit wird eine kanonische Ordnung bestimmt. Der Beweis setzt voraus, dass die Reihenfolge der Knoten auf der Außenfläche von  $G_n$  bekannt ist. Diese Reihenfolge kann vorab in linearer Zeit mit dem Algorithmus aus [31] bestimmt werden. Als nächstes wird eine auf der Außenfläche liegende Kante als Bezugskante  $(v_1, v_2)$  gewählt. Die kanonische Ordnung wird ähnlich wie bei einem ebenen Dreiecksgraph in umgekehrter Reihenfolge bestimmt. Kriterium für die mögliche Elimination eines Knoten während der Berechnung



**Abbildung 48:** Zerlegung in die beiden Bäume  $T_1$  und  $T_2$ .

ist der Grad eines Knotens. Beträgt der Grad zwei, kann der betreffende Knoten entfernt werden. Hilfssatz 14 garantiert, dass ein solcher Knoten  $v_n$  mit  $v_n \neq v_1, v_2$  zu Beginn der Berechnung existiert. Der nach der Entfernung dieses Knotens entstandene Graph ist nach Hilfssatz 13(b) wieder ein maximaler außenplanarer Graph, auf den erneut Hilfssatz 14 angewendet werden kann. Ein weiterer Knoten kann nun entfernt werden. Dies wird solange fortgesetzt, bis nur noch  $v_1$  und  $v_2$  verblieben sind.

Zusammen mit einem Knoten werden die beiden zu ihm inzidenten Kanten entfernt. Durch Vereinbarung seien diese beiden Kanten von ihm weg orientiert. Sie liegen bei ihrer Entfernung auf der Außenfläche des zu diesem Zeitpunkt aktuellen Graphen. Das sukzessive Entfernen der Knoten und Kanten induziert eine kanonische Ordnung der Knoten und eine Einteilung der Kantenmenge in zwei Partitionen. Dabei ist eine Kante zugehörig zur ersten bzw. zur zweiten Partition, wenn sie zum Zeitpunkt ihrer Entfernung auf der Außenfläche in Richtung  $v_1$  bzw. in Richtung  $v_2$  verläuft. In den beiden Partitionen treten keine Zyklen auf. Andernfalls müsste es Kanten geben, die von einem Knoten mit einer kleineren Nummerierung zu einem Knoten mit größerer Nummerierung verlaufen. Dieses ist jedoch aufgrund der Konstruktion nicht der Fall. Die beiden Partitionen stellen daher Bäume dar. Da von  $v_1$  bzw.  $v_2$  keine weiteren Kanten ausgehen, sind dies die Wurzeln für  $T_1$  bzw.  $T_2$ . Damit schließt der Beweis.  $\square$

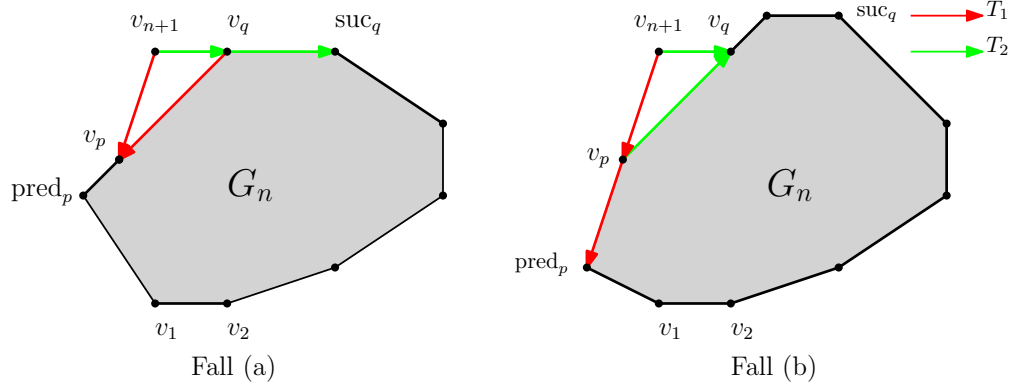
**Hilfssatz 16:** In einem maximalen außenplanaren Graphen  $G_n = (V, E)$  mit  $|V| = n$  haben  $T_1$  und  $T_2$  zusammen  $(n - 1)$  Blätter.

*Beweis.* Der Beweis wird induktiv über die Anzahl der Knoten geführt.

(1) Basisfall: Für  $n = 3$  haben sowohl  $T_1$  als auch  $T_2$  genau ein Blatt. Die Aussage ist also für  $n = 3$  wahr.

(2) Induktionsschluss: Sei die Aussage für  $n$  bewiesen. Ein maximaler außenplanarer Graph  $G_{n+1}$  mit  $(n + 1)$  Punkten entsteht aus einem maximalen außenplanaren Graphen  $G_n$  mit  $n$  Knoten, indem man einen Knoten  $v_{n+1}$  hinzugefügt und ihn mit einer auf der Außenfläche liegenden Kante  $(v_p, v_q)$  verbindet. Der Nachbar von  $v_p$  auf der Außenfläche in Richtung  $v_1$

werde mit  $\text{pred}_p$  bezeichnet, der Nachbar von  $v_q$  auf der Außenfläche in Richtung  $v_2$  mit  $\text{suc}_q$ . Es werden nun zwei Fälle unterschieden:



**Abbildung 49:** Bildung eines neuen Blatts in (a)  $T_1$  beziehungsweise. (b)  $T_2$ .

(a) Die Kante  $(v_p, v_q)$  ist eine  $T_1$ -Kante (Abbildung 49(a)). Somit ist die Kante  $(v_q, \text{suc}_q)$  eine  $T_2$ -Kante. Der Knoten  $v_{n+1}$  ist neues Blatt sowohl in  $T_1$  als auch in  $T_2$ . Da  $(v_q, \text{suc}_q)$  eine  $T_2$ -Kante ist, ist  $v_q$  jedoch nicht mehr Blatt in  $T_2$ . Daher erhöht sich für  $T_2$  die Anzahl der Blätter nicht. Für  $T_1$  hingegen steigt sie um eins.

(b) Die Kante  $(v_p, v_q)$  ist eine  $T_2$ -Kante (Abbildung 49(b)). Somit ist die Kante  $(v_p, \text{pred}_p)$  eine  $T_1$ -Kante. Der Knoten  $v_{n+1}$  ist neues Blatt sowohl in  $T_1$  als auch in  $T_2$ . Da  $(v_p, \text{pred}_p)$  eine  $T_1$ -Kante ist, ist  $v_p$  jedoch nicht mehr Blatt in  $T_1$ . Daher erhöht sich für  $T_1$  die Anzahl der Blätter nicht. Für  $T_2$  hingegen steigt sie um eins.

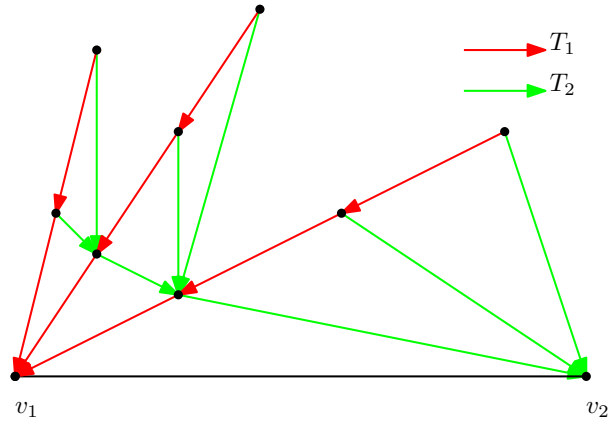
In beiden Fällen hat sich die Summe der Blätter von  $T_1$  und  $T_2$  um eins erhöht, so dass  $T_1$  und  $T_2$  in  $G_{n+1}$  zusammen  $n = (n - 1) + 1$  Blätter haben. Damit ist der Induktionsschluss bewiesen.

(3) Mit Basisfall und Induktionsschluss ist die Behauptung bewiesen.  $\square$

**Definition 17:** In einem maximalen außenplanaren Graphen mit der beschriebenen Zerlegung in  $T_1$  und  $T_2$  wird der Baum mit der geringeren Anzahl von Blättern als Referenzbaum bezeichnet, der andere als komplementärer Baum.

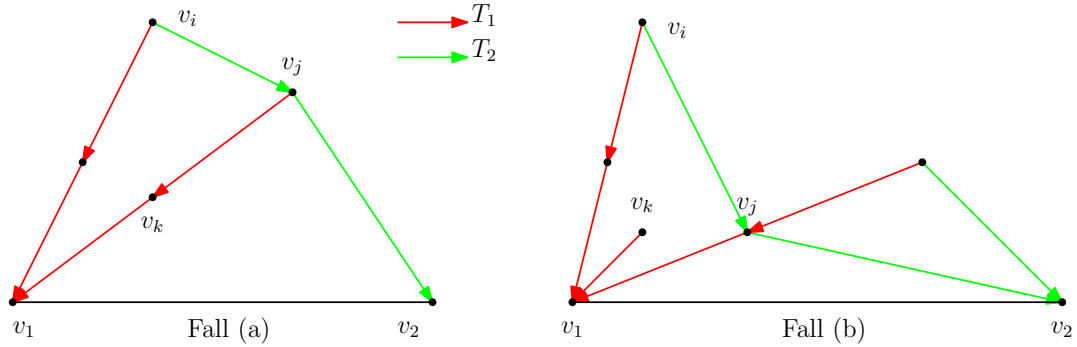
**Hilfssatz 17:** Sei  $G = (V, E)$  ein maximaler außenplanarer Graph,  $T_1$  ( $T_2$ ) der Referenzbaum und  $(v_i, v_j)$  mit  $v_i \neq v_1, v_2$  die  $T_2$ -Kante ( $T_1$ -Kante), die  $v_i$  verlässt. Dann ist  $v_j$  der erste Knoten nach  $v_i$  in einem Preorder-Durchlauf von  $T_1$  ( $T_2$ ) im Uhrzeigersinn (gegen den Uhrzeigersinn), der nicht mit  $v_i$  verwandt ist. Ist  $v_i$  Knoten des letzten Asts, gilt  $v_j = v_2$  ( $v_j = v_1$ ) (Abbildung 50).

*Beweis.* Der Beweis folgt einer Idee aus [3]. Zum Zweck eines Widerspruchsbeweises sei  $v_k$  mit  $k \neq j$  der erste Knoten nach  $v_i$  in einem Preorder-Durchlauf von  $T_1$  ( $T_2$ ) im Uhrzeigersinn (gegen den Uhrzeigersinn), der nicht mit  $v_i$  verwandt ist. Man unterscheidet nun zwei Fälle:



**Abbildung 50:** Nächster nicht verwandter Knoten.

- (a)  $v_k$  ist Vorfahre von  $v_j$ . Dann liegt  $v_k$  im Inneren der Fläche, die durch die Kante  $(v_i, v_j)$ , den Weg  $(v_j, \dots, v_2)$ , der Kante  $(v_2, v_1)$  und den Weg  $(v_1, \dots, v_i)$  gebildet wird. Dies ist ein Widerspruch zu der Tatsache, dass  $G$  außenplanar ist (Abbildung 51(a)).
- (b)  $v_k$  ist nicht Vorfahre von  $v_j$ . Dann liegt  $v_k$  im Inneren der Fläche, die durch die Kante  $(v_i, v_j)$  und den Weg  $(v_j, \dots, v_1, \dots, v_i)$  gebildet wird. Auch dies ist ein Widerspruch zu der Tatsache, dass  $G$  außenplanar ist (Abbildung 51(b)).



**Abbildung 51:** Beweis von Hilfssatz 17.

In beiden möglichen Fällen wurde ein Widerspruch herbeigeführt. Insgesamt wurde die Behauptung also bewiesen.  $\square$

#### 2.4.2. Algorithmus OUTERPLANARFEWSEG

Als nächstes wird Algorithmus OUTERPLANARFEWSEG beschrieben und analysiert, der von einem maximalen außenplanaren Graphen  $G$  eine segmentreduzierte Zeichnung auf einem polynomialen Gitter erstellt.

**Beschreibung:** Zunächst führt Algorithmus OUTERPLANARFEWSEG gemäß Hilfssatz 15 eine Dekomposition von  $G$  in die zwei Bäume  $T_1, T_2$  durch. Ähnlich wie bei Algorithmus 3TREEFEWSEG beruht nun die Strategie zur Reduzierung der Segmentanzahl auf dem Kalkül, einen der durch die Zerlegung erhaltenen Bäume so zu zeichnen, dass ein Blatt



dieses Baums genau ein Segment induziert [14]. Konkret werden für  $G$  entweder die Kanten von  $T_1$  oder die von  $T_2$  so gezeichnet, dass ein Segment, das an einem inneren Knoten dieses Baums beginnt, an einem Blatt des Baums endet. Die Knoten entlang des Pfades von einem inneren Knoten zu einem Blatt werden daher mit kollinearen Punkten gezeichnet. Die Anzahl der Segmente eines so gezeichneten Baums ist dann gleich der Anzahl seiner Blätter. Im Einzelnen besteht folgender Ablauf:

Zunächst wird mit dem Algorithmus aus [31] die Reihenfolge der Knoten auf der Außenfläche von  $G$  berechnet. Dann wird die Bezugskante  $(v_1, v_2)$  gewählt. Anschließend werden die beiden Bäume  $T_1$  und  $T_2$  gemäß Hilfssatz 15 mit Bezug auf  $(v_1, v_2)$  berechnet. Nach Berechnung der Anzahl der Blätter in jedem Baum werden Referenzbaum und komplementärer Baum bestimmt. In Abbildung 48 zum Beispiel ist  $T_1$  der Referenzbaum mit sechs Blättern, während  $T_2$  der komplementäre Baum mit sieben Blättern ist. Ziel ist es, die Kanten des Referenzbaums entsprechend dem vorstehend beschriebenen Kalkül zu zeichnen und so eine Reduzierung der Segmentanzahl zu erreichen.

Ohne Beschränkung der Allgemeinheit gehen die weiteren Ausführungen davon aus, dass  $T_1$  der Referenzbaum ist. Ist er dies in tatsächlicher Hinsicht nicht, benennt man die beiden Bäume um und spiegelt zum Schluss noch die Zeichnung.

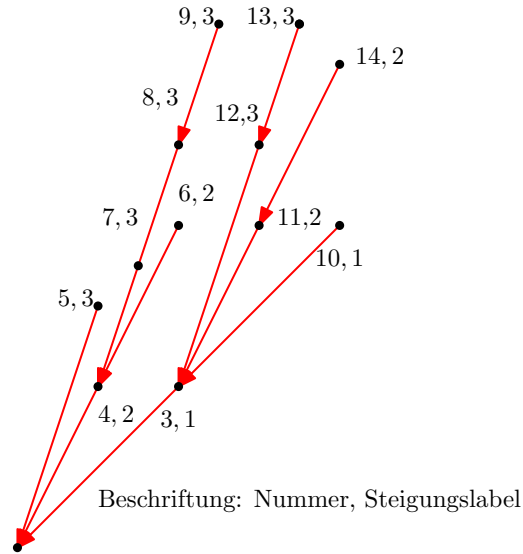
Insgesamt sind zwei Preorder-Durchläufe durch den Referenzbaum und ein Durchlauf durch den komplementären Baum notwendig. Der erste der beiden Durchläufe durch den Referenzbaum wird gegen den Uhrzeigersinn durchgeführt, der zweite im Uhrzeigersinn. Für die Traversierung des komplementären Baums besteht keine vorgeschriebene Richtung.

Im ersten Durchlauf durch den Referenzbaum wird den Kanten des Referenzbaums die Steigung zugewiesen, mit der sie gezeichnet werden. Im zweiten Durchlauf durch den Referenzbaum werden dann sukzessive alle Knoten und alle Kanten des Referenzbaums gezeichnet. Während der anschließenden Traversierung des komplementären Baums werden die Kanten des komplementären Baums gezeichnet.

Für die Zuweisung der Kantensteigung erhält jeder Knoten ein Steigungsetikett, in dem die Steigung der von ihm zu seinem Elternknoten verlaufenden Kante gespeichert wird. Die Ermittlung der Steigung findet auf folgende Weise statt:

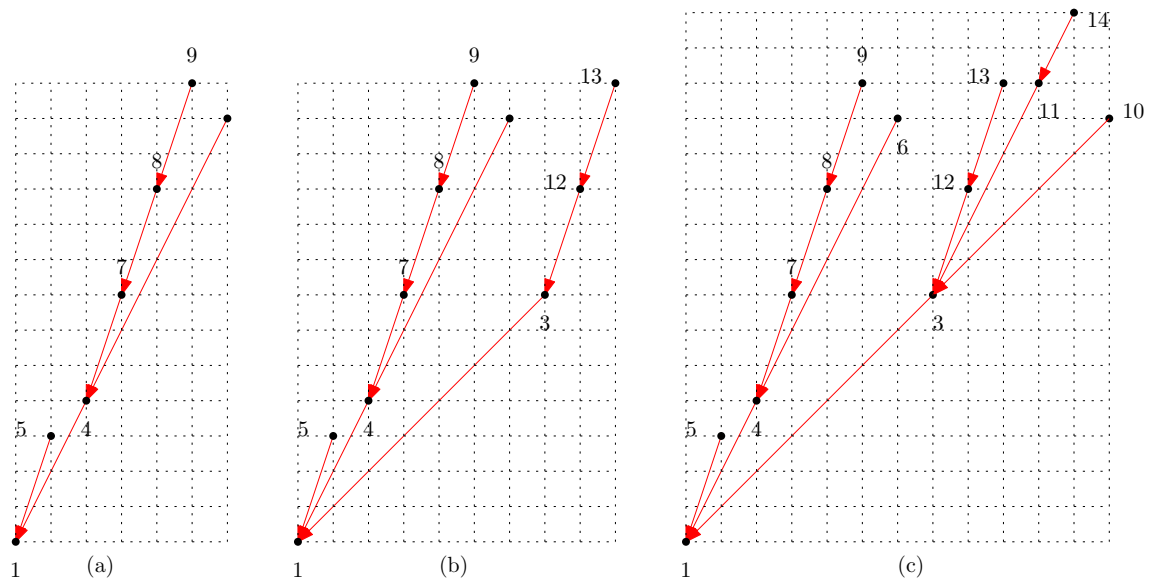
- (1) Das Steigungsetikett von  $v_3$  hat den Wert eins. Dies bedeutet, dass die Kante  $(v_1, v_3)$  mit der Steigung eins gezeichnet wird.
- (2) Das erste zu zeichnende Kind eines Knotens erhält das gleiche Steigungsetikett wie der Knoten selbst. Dies bedeutet, dass die Kante von dem ersten zu zeichnenden Kind zu seinem Elternknoten dieselbe Steigung hat wie die vom Elternknoten ausgehende Kante. Hierdurch wird die Strategie zur Reduzierung der Segmentanzahl realisiert.
- (3) Jedes nächste zu zeichnende Kind eines Knotens erhält ein im Wert um eins erhöhtes Steigungsetikett.

Damit ist die erste Traversierung des Referenzbaums abgeschlossen. Abbildung 52 zeigt den Referenzbaum aus Abbildung 48 mit zusätzlich annotierten Steigungsetiketten.



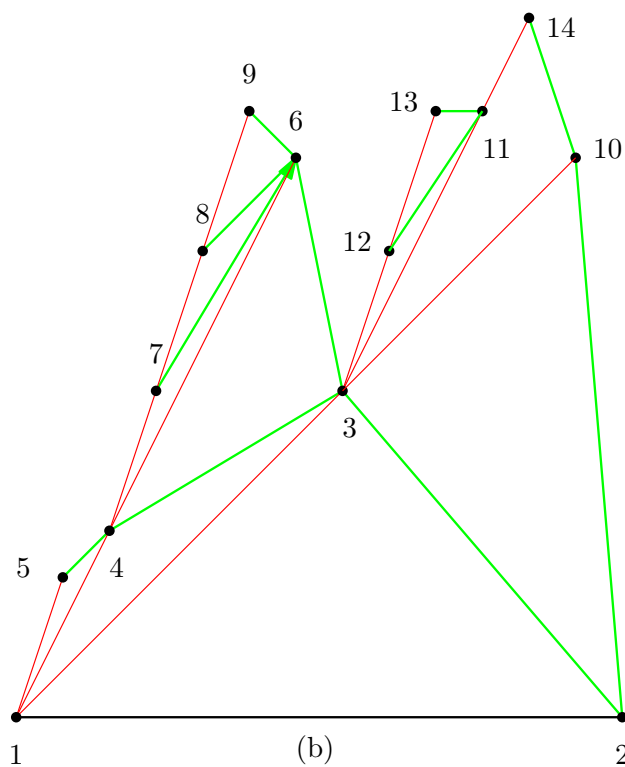
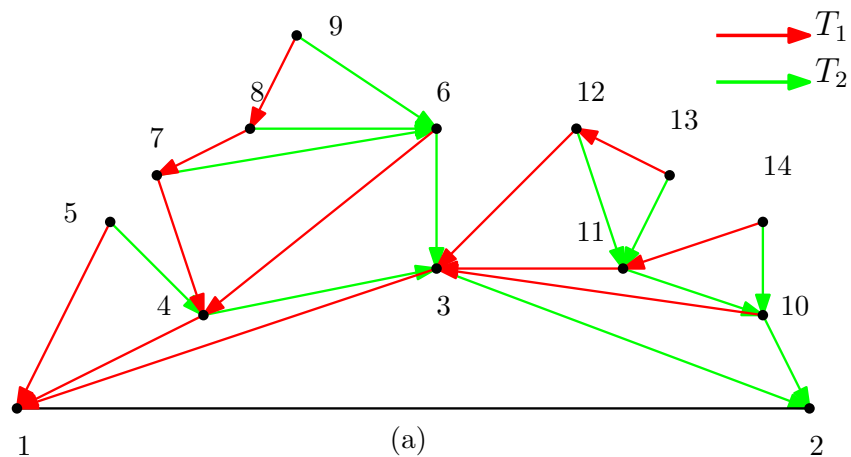
**Abbildung 52:** Steigungslabels.

Während der zweiten Traversierung findet die Zeichnung des Referenzbaums statt. Vorbereitend wird  $v_1$  mit den Koordinaten  $(0,0)$  und  $v_2$  mit den Koordinaten  $(n,0)$  gezeichnet. Dann startet bei  $v_1$  ein Preorder-Durchlauf von  $T_1$  im Uhrzeigersinn. Während dieses Durchlaufs wird die  $x$ -Koordinate für die Zeichnung des nächsten Knotens jeweils um eins erhöht (erniedrigt). Im Steigungslabel des nächsten zu zeichnenden Knotens ist die Steigung der Kante von diesem Knoten zu seinem Elternknoten gespeichert. Aus diesem Label ergibt sich dann in Verbindung mit der bereits ermittelten neuen  $x$ -Koordinate auch die  $y$ -Koordinate für die Zeichnung des nächsten Knotens. Abbildung 53 zeigt in verschiedenen Stadien den auf diese Weise gezeichneten Referenzbaum  $T_1$  aus Abbildung 52.



**Abbildung 53:** Gezeichneter Referenzbaum  $T_1$  in verschiedenen Stadien.

Mit einem Tiefendurchlauf durch den komplementären Baum wird die Zeichnung um die Kanten des komplementären Baums ergänzt. In Abbildung 54(a) ist noch einmal die ursprüngliche Zeichnung von  $G$  einschließlich seiner Zerlegung zu sehen, während Abbildung 54(b) die fertige segmentreduzierte Zeichnung von  $G$  zeigt.



**Abbildung 54:** (a) Maximaler außenplanarer Graph  $G$  und (b) seine segmentreduzierte Zeichnung.

Die Eigenschaften von Algorithmus OUTERPLANARFEWSEG werden im folgenden Satz 5 beschrieben und zusammengefasst:

**Satz 5:** Algorithmus OUTERPLANARFEWSEG erstellt von einem maximalen außenplanaren Graphen  $G = (V, E)$  mit  $|V| = n$ ,  $|E| = e$  eine Zeichnung  $\Gamma$  auf dem Gitter mit kreuzungsfreien geraden Kanten. Dabei hat  $\Gamma$  eine visuelle Komplexität von  $s = \frac{3}{2}n - \frac{3}{2}$  beziehungsweise  $s = \frac{3}{4}e + O(1)$ . Die Gittergröße beträgt  $O(n) \times O(n^2)$ ; diese Schranke ist eng.

*Beweis.* Offensichtlich erstellt Algorithmus OUTERPLANARFEWSEG von  $G$  eine Gitterzeichnung mit geraden Kanten. Es erfolgt nun der Nachweis, dass diese Gitterzeichnung kreuzungsfrei ist. Danach werden visuelle Komplexität und Gitterschranke berechnet:

**Kreuzungsfreiheit:** Der Nachweis der Kreuzungsfreiheit geschieht getrennt für die Kanten des Referenzbaums und des komplementären Baums. Bei der Zeichnung der Kanten des Referenzbaums unterscheidet man zwei Fälle:

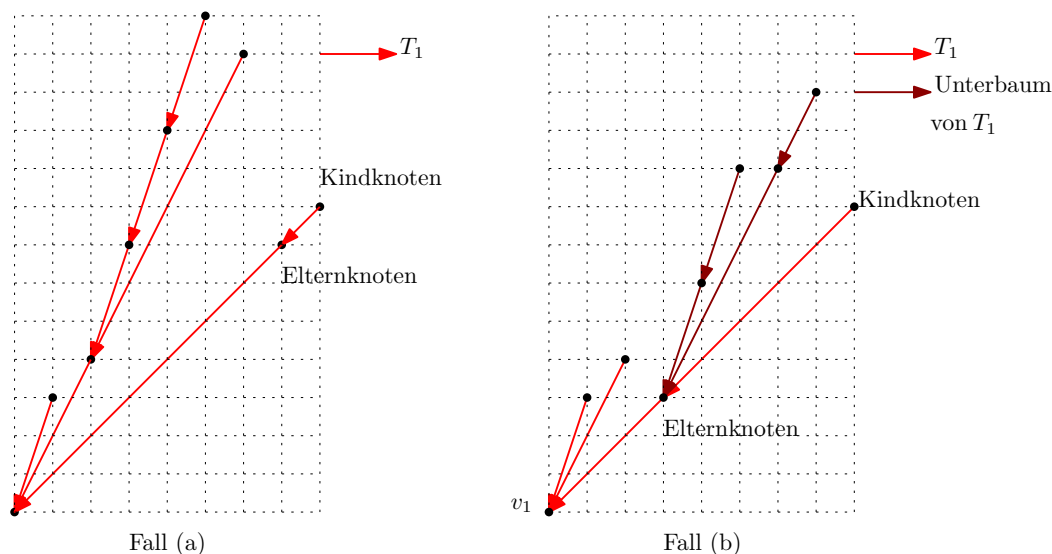
- (1) Der nächste zu zeichnende Knoten ist das erste Kind seines Elternknotens, das gezeichnet wird.

Aufgrund der Reihenfolge der zu besuchenden Knoten hat der Elternknoten die größte  $x$ -Koordinate im bisher gezeichneten Teil von  $T_1$ . Da der zu zeichnende Kindknoten eine bezogen auf den Elternknoten um eins erhöhte  $x$ -Koordinate erhält, kann die zu zeichnende Verbindungskante zwischen Eltern- und Kindknoten keine andere bisher gezeichnete Kante kreuzen (Abbildung 55(a)). Somit kann in diesem Fall der nächste Knoten gezeichnet werden, ohne kreuzende Kanten zu induzieren.

- (2) Der nächste zu zeichnende Knoten ist nicht das erste Kind seines Elternknotens, das gezeichnet wird.

Genau die bisher gezeichneten Knoten des an dem Elternknoten verwurzelten Unterbaums haben eine größere  $x$ -Koordinate als der Elternknoten. Der nächste Knoten wird mit einer  $x$ -Koordinate gezeichnet, die um eins größer ist als die größte  $x$ -Koordinate eines bisher gezeichneten Knotens in diesem Unterbaum. Daher könnten genau die bisher gezeichneten Kanten dieses Unterbaums die Verbindungskante zwischen dem nächsten zu zeichnenden Knoten und dem Elternknoten kreuzen. Da dieser Kante im vorbereitenden Durchlauf jedoch eine Steigung zugewiesen worden ist, die kleiner ist als alle Steigungen der bisher gezeichneten Kanten dieses Unterbaums, kann eine Kreuzung zwischen dieser Verbindungskante und den bisher gezeichneten Kanten des Unterbaums ausgeschlossen werden (Abbildung 55(b)). Somit kann auch in diesem Fall der nächste Knoten gezeichnet werden, ohne kreuzende Kanten zu induzieren.

Es wurde bewiesen, dass sukzessive alle Knoten und Kanten des Referenzbaums gezeichnet werden können, ohne Kreuzungen zu induzieren. Insgesamt wird also der Referenzbaum kreuzungsfrei gezeichnet.



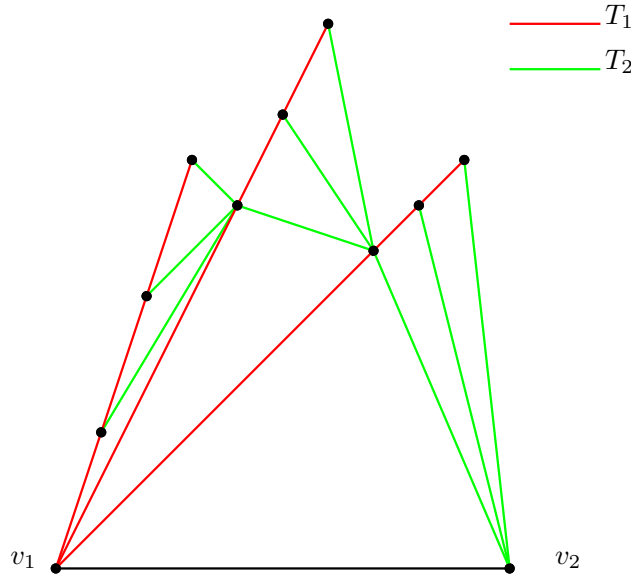
**Abbildung 55:** Kreuzungsfreiheit der Kanten des Referenzbaums.

Es verbleibt der Nachweis, dass die Kanten des komplementären Baums eingefügt werden können, ohne Kreuzungen hervorzurufen. Nach Hilfssatz 17 verlaufen die Kanten des komplementären Baums von einem Knoten zu dessen nächsten nicht verwandten Knoten—bezogen auf einen Preorder-Durchlauf im Uhrzeigersinn durch den Referenzbaum. Im Folgenden wird der Knoten, den die Kante des komplementären Baums verlässt, als Ursprungsknoten der Kante bezeichnet; entsprechend wird der zweite zu dieser Kante inzidente Knoten als Zielknoten bezeichnet. Im Sinne einer vereinheitlichenden Betrachtungsweise wird die Kante  $(v_1, v_2)$  als zugehörig zum Referenzbaum angesehen. Es gilt (Abbildung 56):

- (1) Der Ursprungsknoten einer Kante des komplementären Baums ist im Referenzbaum durch ein gerades—gegebenenfalls aus mehreren Kanten bestehendes—Segment mit dem Elternknoten des Zielknotens verbunden.
- (2) Die am Elternknoten auf dieses Segment im Uhrzeigersinn folgende Kante des Referenzbaums verläuft zum Zielknoten.
- (3) In dem durch den Ursprungsknoten, den Elternknoten des Zielknotens und dem Zielknoten gebildeten Dreieck kann daher keine weitere Kante des Referenzbaums liegen.

Daraus folgt, dass eine Kante des komplementären Baums eine Kante des Referenzbaums nicht kreuzen kann. Es verbleibt der Nachweis, dass sich die Kanten des komplementären Baums nicht untereinander kreuzen können. Haben Kanten des komplementären Baums denselben Zielknoten, liegen ihre Ursprungsknoten nacheinander auf demselben geraden Segment des Referenzbaums. Die so zueinander liegenden Kanten des komplementären Baums bilden eine Gruppe. Die Kanten einer Gruppe können sich offenbar nicht gegenseitig kreuzen. Die einzelnen Gruppen wiederum werden durch die Segmente des Referenzbaums voneinander abgegrenzt. Daher können sich Kanten des komplementären Baums, die verschiedenen Gruppen angehören, nicht gegenseitig kreuzen. Damit ist gezeigt, dass die Kanten des komplementären Baums keine Kreuzungen verursachen. Insgesamt ist daher nachgewiesen,

dass Algorithmus OUTERPLANARFEWSEG eine Zeichnung mit kreuzungsfreien, geradlinigen Kanten anfertigt.



**Abbildung 56:** Kreuzungsfreiheit der Kanten des komplementären Baums.

**Visuelle Komplexität:** Die Gesamtanzahl  $s$  der Segmente ist gleich der Summe der Segmente, mit denen der Referenzbaum und der komplementäre Baum gezeichnet werden. Hinzu kommt ein Segment für die Zeichnung der Bezugskante. Die Anzahl der Segmente des Referenzbaums ist konstruktionsbedingt gleich der Anzahl seiner Blätter, die Anzahl der Segmente des komplementären Baums ist höchstens gleich der Anzahl seiner Kanten, also höchstens gleich  $(n - 2)$ . Der Referenzbaum wurde so gewählt, dass die Anzahl seiner Blätter kleiner (oder gleich) der Anzahl der Blätter des komplementären Baums ist. Mit Hilfssatz 16 ist die Anzahl der Blätter im Referenzbaum also höchstens gleich  $\lfloor \frac{n-1}{2} \rfloor$ . Insgesamt ergibt sich also für die visuelle Komplexität:

$$\begin{aligned} s &\leq \frac{n-1}{2} + (n-2) + 1 \\ s &\leq \frac{3n-3}{2} \\ s &\leq \frac{3}{2}n - \frac{3}{2} \end{aligned}$$

Da jeder maximale außenplanare Graph  $e = 2n - 3$  Kanten hat, folgt weiterhin:

$$s \leq \frac{3}{4}e + O(1)$$

**Gitterschranke:** Die Breite der Zeichnung ist gleich  $n$ . Mit  $T_1$  bzw.  $T_2$  als Referenzbaum ist die größte bzw. kleinste Steigung einer Kante des Referenzbaums gleich  $(n-2)$  bzw. gleich  $(-n+2)$ . In Verbindung mit der Breite von  $n$  Längeneinheiten ist daher die asymptotische Schranke für die Höhe gleich  $O(n^2)$ . Diese Schranke ist scharf, wie die im Folgenden definierte Klasse  $G_{4k}$ ,  $k \in \mathbb{N}$  von maximal außenplanaren Graphen zeigt:

- Abbildung 57 zeigt den Referenzbaum für  $k = 2$  und die von Algorithmus OUTERPLANAR-FEWSEG erzeugte Zeichnung des Referenzbaums. Der zweite Knoten des mittleren Asts, also des  $k$ -ten Asts, hat die  $x$ -Koordinate  $2k$ , während die Kante von  $v_1$  zu diesem Knoten wird mit einer Steigung von  $k$  gezeichnet. Die  $y$ -Koordinate dieses Knotens ist daher  $2k^2$ . Damit ist nachgewiesen, dass die Gitterschranke scharf ist. Dies vervollständigt den Beweis von Satz 5.  $\square$



73

## 2.5. Segmentreduzierte Gitterzeichnungen von serien-parallelen Graphen

Mit ihrer Möglichkeit, sowohl serielle als auch parallele Abläufe beschreiben zu können, eignen sich serien-parallele Graphen insbesondere für Ablaufdiagramme und für die Netzplantechnik. Historisch hatten sie ihre ersten Anwendungen in der Modellierung von elektrischen Schaltkreisen. Sie haben zwei ausgezeichnete Knoten, die Quelle und die Senke. Auch in einem ungerichteten serien-parallelen Graphen kann den Kanten eine Orientierung zugeordnet werden. Die Quelle ist dann der einzige Knoten, der nur ausgehende Kanten hat. Die Senke dagegen ist der einzige Knoten, der nur eingehende Kanten hat. Zunächst beschränken sich die Ausführungen auf 2-zusammenhängende serien-parallele Graphen, bevor am Ende des Kapitels die Ergebnisse auch auf nicht 2-zusammenhängende serien-parallele Graphen übertragen werden. Für serien-parallele Graphen mit Grad drei wird ein separater Algorithmus beschrieben. Der folgende Grundlagenabschnitt beginnt mit der formalen Definition von serien-parallelen Graphen und der Beschreibung der Dekompositionstechnik der Ohrenzerlegung.

### 2.5.1. Spezifische Grundlagen

**Definition 18 ([2]):** Ein serien-paralleler Graph wird rekursiv wie folgt definiert:

Eine Kante, die zwei Knoten miteinander verbindet, ist ein serien-paralleler Graph. Die beiden Knoten werden Quelle bzw. Senke genannt. Seien  $G_1$  und  $G_2$  zwei serien-parallele Graphen. Dann stellt sowohl die serielle als auch die parallele Verkettung von  $G_1$  und  $G_2$  einen serien-parallelen Graphen dar. Die serielle Verkettung von  $G_1$  und  $G_2$  ist der Graph, den man durch Gleichsetzen der Senke von  $G_1$  mit der Quelle von  $G_2$  erhält. Die parallele Verkettung von  $G_1$  und  $G_2$  erhält man, indem man die Quelle von  $G_1$  mit der Quelle von  $G_2$  und die Senke von  $G_1$  mit der Senke von  $G_2$  gleichsetzt.

**Definition 19 ([15]):** Die Dekompositionstechnik der *Ohrenzerlegung* lässt sich folgendermaßen beschreiben:

- (1) Eine Ohrenzerlegung eines ungerichteten Graphen  $G$  ist definiert als eine Einteilung der Kanten von  $G$  in eine Sequenz von *Ohren*  $E_1, E_2, \dots, E_n$ . Jedes Ohr ist ein Weg im Graph mit folgenden Eigenschaften:
  - (a) Wenn ein Knoten mehrfach auf dem Weg liegt, dann ist er Anfangs- und Endknoten.
  - (b) Die beiden Endpunkte jedes Ohrs  $E_i, i > 1$  erscheinen in früheren Ohren  $E_j$  und  $E_{j'}$ , mit  $j < i$  und  $j' < i$ .
  - (c) Kein innerer Punkt von  $E_i$  liegt in einem Ohr  $E_j$  mit  $i > j$ .
  - (d) Typischerweise bestehen weitere Einschränkungen für das erste Ohr. So kann zum Beispiel gefordert werden, dass es ein einzelner Knoten oder eine einzelne Kante ist.
- (2) Eine offene Ohrenzerlegung ist eine Ohrenzerlegung, in der die beiden Endpunkte von jedem Weg unterschiedlich sind. Dies bedeutet, dass kein Weg einen Kreis darstellt. Man sagt, eine offene Ohrenzerlegung startet von einem Weg  $P$  aus, wenn  $E_1 = P$  ist.



- (3) Sei ein Graph  $G$  und eine offene Ohrenzerlegung  $\mathcal{D} = (E_1, E_2, \dots, E_n)$  von  $G$  gegeben. Man sagt, dass  $E_i$  in  $E_j$  verschachtelt ist, wenn  $j < i$  gilt und wenn beide Endpunkte von  $E_i$  in  $E_j$  erscheinen. Für solche  $i$  und  $j$  wird als Schachtelintervall von  $E_i$  in  $E_j$  der Weg in  $E_j$  zwischen den Endpunkten von  $E_i$  definiert.  $\mathcal{D}$  ist eine verschachtelte offene Ohrenzerlegung, wenn die folgenden beiden Bedingungen erfüllt sind:
- (a) Für jedes  $i > 1$  gibt es ein  $j < i$ , so dass  $E_i$  in  $E_j$  verschachtelt ist.
  - (b) Wenn zwei Ohren  $E_i$  und  $E_{i'}$  beide in demselben Ohr  $E_j$  verschachtelt sind, dann enthält entweder das Schachtelintervall von  $E_i$  in  $E_j$  das Schachtelintervall von  $E_{i'}$  in  $E_j$  oder umgekehrt, oder aber die beiden Schachtelintervalle sind disjunkt. Dies bedeutet, dass sich in einem beliebigen Ohr  $E_i$  keine zwei Schachtelintervalle überkreuzen.

**Definition 20 ([15]):** Es werden die beiden Begriffe „richtig verschachtelt“ und „enthalten“ unterschieden. Der Begriff „richtig verschachtelt“ entspricht dem Verschachtelungsbegriff aus Definition 19. Dagegen ist ein Ohr  $E_i$  enthalten in  $E_j$ , wenn es eine Folge von Ohren  $E_{j,0}, \dots, E_{j,m}$  gibt, für die gilt:  $E_i$  ist richtig verschachtelt in  $E_{j,0}$ ,  $E_{j,k}$  ist richtig verschachtelt in  $E_{j,(k+1)}$  für  $0 \leq k < m$  und  $E_{j,m}$  ist richtig verschachtelt in  $E_j$ .

**Definition 21 ([11]):** Der Begriff der kombinatorischen Isomorphie wird folgendermaßen definiert:

Sei  $G = (V, E)$  ein Graph und  $G' = (V', E')$  ein weiterer Graph. Sei  $F$  die Menge aller Gebiete von  $G$  und  $F'$  die Menge aller Gebiete von  $G'$ .  $G$  heißt kombinatorisch isomorph zu  $G'$ , wenn es eine Bijektion  $\sigma : V \cup E \cup F \rightarrow V' \cup E' \cup F'$  gibt, die folgende Eigenschaften hat:

- (1) Durch  $\sigma$  wird  $V$  auf  $V'$  und  $E$  auf  $E'$  abgebildet.
- (2) Durch  $\sigma$  werden die Inzidenzen von Ecken mit Kanten erhalten.
- (3) Durch  $\sigma$  werden die Inzidenzen von Ecken und Kanten mit Gebieten erhalten.

**Definition 22:** Ein Ohr, das nur eine Kante hat, wird als triviales Ohr bezeichnet.

**Definition 23:** Jedem nichttrivialen Ohr  $E_i \neq E_1$  wird eine Krümmung wie folgt zugeordnet: Sei  $E_j$  das Ohr, in dem  $E_i$  verschachtelt ist. Man betrachte den Kreis, der entsteht, wenn man am Anfangsknoten von  $E_i$  startet, in  $E_i$  entlangschreitet bis zum Endknoten von  $E_i$  und dann in  $E_j$  zurück zum Anfangsknoten von  $E_i$  schreitet. Dann wird dem Ohr die Krümmung dieses Kreises zugeordnet.

**Hilfssatz 18 ([15]):** Ein ungerichteter Graph  $G$  ist genau dann ein serien-paralleler Graph mit Quelle  $s$  und Senke  $t$ , wenn er eine verschachtelte offene Ohrenzerlegung hat, die von einem Pfad startet, der in  $s$  beginnt und in  $t$  endet.

Der Hilfssatz 18 definiert somit einen serien-parallelen Graphen über seine verschachtelte offene Ohrenzerlegung, während in dem folgenden Hilfssatz der Zusammenhang zwischen einer solchen Zerlegung und einer offenen Ohrenzerlegung hergestellt wird.

**Hilfssatz 19 ([15]):** Eine offene Ohrenzerlegung eines serien-parallelen Graphen ist eine verschachtelte offene Ohrenzerlegung.

**Hilfssatz 20 ([15]):** Für die Quelle  $s$  und die Senke  $t$  und damit auch für die Wahl des Weges, von dem die verschachtelte offene Ohrenzerlegung aus startet, gilt:

- (1) Ist  $G$  ein 2-zusammenhängender serien-paralleler Graph und  $(s, t)$  eine beliebige Kante in  $G$ , dann ist  $G$  ein serien-paralleler Graph mit Quelle  $s$  und Senke  $t$ .
- (2) Ist  $G$  ein nicht 2-zusammenhängender serien-paralleler Graph, so gilt:
  - (a) Der Baum der 2-zusammenhängenden Komponenten von  $G$  (*block cut vertex tree nach [22]*) ist ein nicht trivialer Weg.
  - (b) Seien  $X$  und  $Y$  die 2-zusammenhängenden Komponenten an den beiden Endpunkten dieses Weges und  $x$  bzw.  $y$  der Artikulationsknoten zwischen  $X$  bzw.  $Y$  und dem Rest des Graphen. Ferner sei  $s$  ein beliebiger Knoten in  $X$ , der benachbart zu  $x$  ist, und  $t$  ein beliebiger Knoten in  $Y$ , der benachbart zu  $y$  ist. Dann ist  $G$  ein serien-paralleler Graph mit Quelle  $s$  und Senke  $t$ .

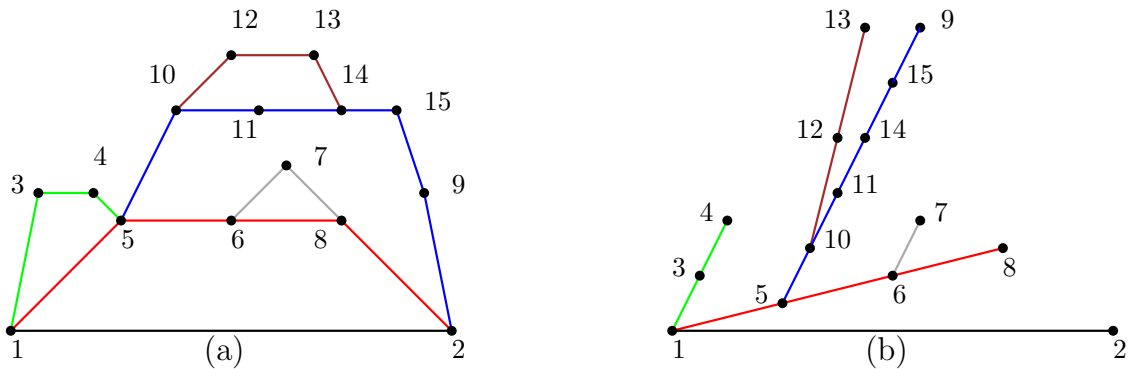
**Hilfssatz 21 ([15]):** In einer verschachtelten offenen Ohrenzerlegung lässt sich für jedes Ohr  $E_i$  die Verschachtelungsstruktur der in  $E_i$  richtig verschachtelten Ohren durch einen Baum beschreiben, den sogenannten Verschachtelungsbaum von  $E_i$ . Mittelbar ergeben alle Verschachtelungsbäume zusammengenommen den *Dekompositionsbaum* des serien-parallelen Graphen, der die Zerlegung des Graphen in serielle und parallele Verkettungen beschreibt.

**Definition 24 ([2]):** Der Dekompositionsbaum wird auch als *SPQ-Baum* bezeichnet. In seiner binären Form beschreibt er direkt, aus welchen seriellen und parallelen Verkettungsoperationen der serien-parallele Graph zusammengesetzt ist. Ein S-Knoten beschreibt eine serielle Verkettung, ein P-Knoten eine parallele Verkettung. Ein Q-Knoten ist das Terminalsymbol und beschreibt eine Kante im serien-parallelen Graphen. Konsekutive serielle Operationen oder konsekutive parallele Operationen werden häufig in einem einzigen S-Knoten beziehungsweise einem einzigen P-Knoten zusammengefasst. In einem solchen komprimierten, nun nicht mehr binären SPQ-Baum kann ein S-Knoten keinen S-Knoten und ein P-Knoten keinen P-Knoten als direkten Nachkommen haben.

### 2.5.2. Beschreibung von Algorithmus SPFEWSEG

Algorithmus SPFEWSEG erstellt von einem 2-zusammenhängenden serien-parallelen Graphen  $G$  eine segmentreduzierte Gitterzeichnung. In diesem Kapitel wird zunächst sein Verlauf und der von ihm verwendete Kalkül zur Verringerung der Segmentanzahl beschrieben. Anschließend wird der Nachweis der Kreuzungsfreiheit seiner Kanten geführt.

**Verlauf von Algorithmus SPFEWSEG:** Es werde vorausgesetzt, dass  $G$  keine Schlingen und keine Multikanten hat. Die grundlegende Idee dieses Algorithmus ist, für  $G$  zunächst eine verschachtelte offene Ohrenzerlegung zu erzeugen, deren Ohren in  $s$ - $t$ -Richtung orientiert sind. Anschließend wird dann aus jedem Ohr (mit Ausnahme des ersten) die jeweils letzte Kante entfernt. Die so verbliebenen Kanten stellen dann zusammen mit den Knoten einen Spannbaum dar, der in der Quelle verwurzelt ist (Abbildung 58).

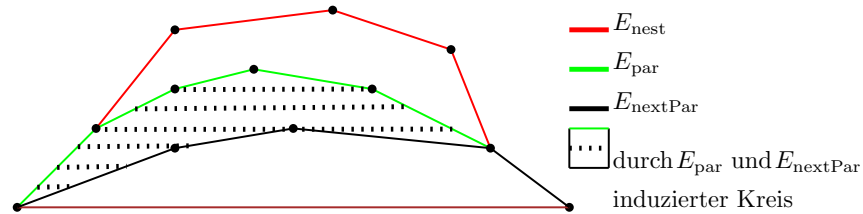


**Abbildung 58:** (a) Ein serien-paralleler Graph  $G$  mit verschachtelter offener Ohrenzerlegung. (b) Spannbaum von  $G$ .

Im Verlauf einer Traversierung des Spannbaums wird dann eine Zeichnung des Graphen erstellt. Die Strategie zur Verringerung der Segmentanzahl ist das bekannte Kalkül, den Spannbaum mit der Anzahl von Segmenten zu zeichnen, die der Anzahl seiner Blätter entspricht. Die Zuweisungsstrategie für die Steigung eines Segmentes besteht darin, bei Beginn eines neuen Asts die Steigung um eins zu erhöhen. Die Zeichenfolge der Knoten entspricht dabei nicht der eines üblichen Preorder-Durchlaufs durch den Spannbaum. Diejenigen Knoten, die Endpunkt (mindestens) eines Ohrs und damit Zielknoten (mindestens) einer letzten Kante sind, werden nicht unmittelbar nach Zeichnung ihres Elternknotens im Spannbaum gezeichnet. Vielmehr werden sie dann gezeichnet, wenn der letzte von den Knoten gezeichnet ist, die Anfangsknoten einer zu dem besagten Knoten inzidenten letzten Kante sind. Dies ist neben der Zuweisungsstrategie für die Steigungen das zentrale Designprinzip des Algorithmus. Die Zuweisungsvorschrift für die Steigungen und die beschriebene Zeichenfolge stimmen im Ergebnis mit dem Vorgehen bei der Erstellung von monotonen Zeichnungen serien-paralleler Graphen überein [24]. Im Einzelnen liegt folgender Ablauf vor:

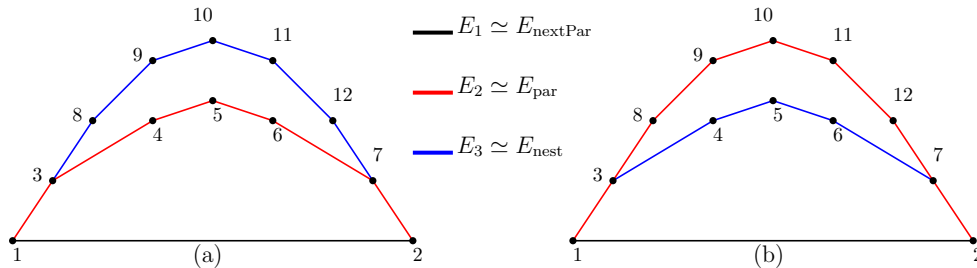
Sei zunächst angenommen, dass die Kante  $(s, t)$  sich gegen den Uhrzeigersinn auf dem Rand der Außenfläche befindet. Das Konzept wird später so erweitert, dass Wahlfreiheit für  $(s, t)$  besteht. Sei ferner eine verschachtelte offene Ohrenzerlegung von  $G$  gegeben, deren Ohren in  $s$ - $t$ -Richtung orientiert sind. Die Ohren mit einem gemeinsamen Anfangsknoten  $v_i$  werden entsprechend der kombinatorischen Reihenfolge ihrer ersten Kanten um  $v_i$  herum angeordnet. Wie beschrieben werden nun die letzten Kanten der Ohren (mit Ausnahme des ersten Ohrs) entfernt. Da der Endknoten eines Ohrs gleichzeitig interner Knoten des Ohrs ist, in dem das betrachtete Ohr verschachtelt ist, ergeben die verbliebenen Kanten den erwähnten Spannbaum von  $G$ .

Sei  $E_{\text{nest}}$  ein Ohr, das in  $E_{\text{par}}$  mit  $E_{\text{par}} \neq E_1$  verschachtelt sei. Mit  $E_{\text{nextPar}}$  sei das Ohr bezeichnet, in dem wiederum  $E_{\text{par}}$  verschachtelt ist. Weiterhin unter Annahme, dass  $(s, t)$  gegen den Uhrzeigersinn auf der Außenfläche liegt, wird es sich für die Korrektheit von Algorithmus SPFEWSEG als notwendig erweisen, dass  $E_{\text{nest}}$  außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verläuft (Abbildung 59).



**Abbildung 59:** Terminologie der Verschachtelungsstruktur.

Wenn eine von  $(s, t)$  startende verschachtelte offene Ohrenzerlegung diese Eigenschaft erfüllt, wird sie im Weiteren als *ansprechende verschachtelte offene Ohrenzerlegung* bezeichnet.<sup>7</sup> Eine ansprechende verschachtelte offene Ohrenzerlegung entspricht der intuitiven Zerlegung eines serien-parallelen Graphen. In Abbildung 60(a) wird ein serien-paralleler Graph zusammen mit einer ansprechenden verschachtelten offenen Ohrenzerlegung gezeigt, während Abbildung 60(b) denselben Graphen mit einer beliebigen verschachtelten offenen Ohrenzerlegung zeigt.

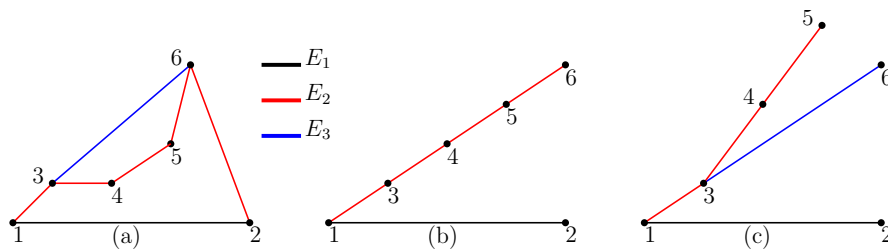


**Abbildung 60:** (a) Serien-paralleler Graph  $G$  mit ansprechender verschachtelter offener Ohrenzerlegung. (b)  $G$  mit beliebiger verschachtelter offener Ohrenzerlegung.

<sup>7</sup>Für 3-zusammenhängende Graphen ist ein solcher Begriff nicht wohldefiniert.

Die eigentliche Zeichnung wird während eines Durchlaufs durch den Spannbaum gegen den Uhrzeigersinn berechnet. Wenn die Konstruktion des Spannbaums auf einer ansprechenden verschachtelten offenen Ohrenzerlegung beruht, wird daher der Elternknoten eines Zielknotens einer letzten Kante grundsätzlich vor dem Anfangsknoten dieser letzten Kante gezeichnet. Diese Eigenschaft wird in der weiteren Beweisführung verwendet.

Im Folgenden wird vorausgesetzt, dass der Spannbaum auf der Basis einer ansprechenden verschachtelten offenen Ohrenzerlegung konstruiert worden ist. Wie man eine solche Zerlegung erhält, wird später beschrieben. Vor der Traversierung des Spannbaums müssen noch die trivialen Ohren besonders berücksichtigt werden. Die (einzige) Kante eines trivialen Ohrs gehört zu den letzten Kanten und wird daher nicht als Kante des Spannbaums gezeichnet. Aufgrund der Eigenschaften einer verschachtelten offenen Ohrenzerlegung liegen Anfangs- und Endpunkt dieses trivialen Ohrs auf demselben Ast des Spannbaums. In der angestrebten Zeichnung ist die Kante des trivialen Ohrs daher deckungsgleich mit dem geraden Kantenzug zwischen ihrem Anfangs- und Endpunkt (Abbildung 61(a) und (b)) und somit als Kante nicht zu identifizieren. Um eine korrekte Zeichnung zu erzielen, muss daher der Spannbaum geringfügig modifiziert werden. Dadurch ändert sich die Einbettung des Graphen.

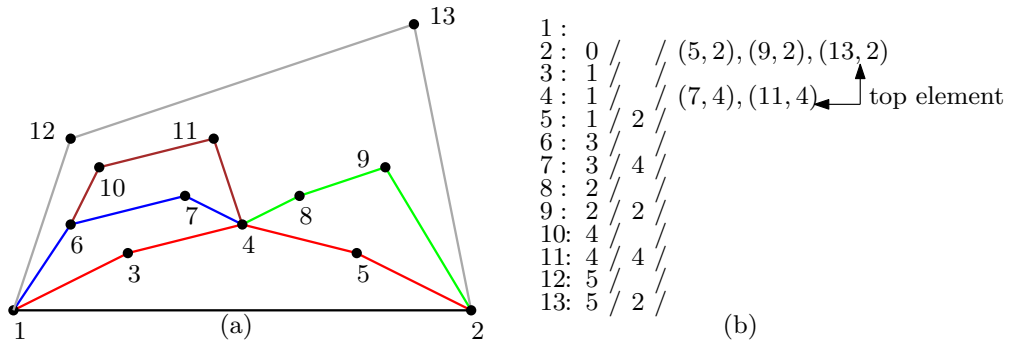


**Abbildung 61:** (a) Serien-paralleler Graph  $G$  mit einem trivialen Ohr. (b) Spannbaum von  $G$ . (c) modifizierter Spannbaum von  $G$ .

Die trivialen Ohren einer ansprechenden verschachtelten offenen Ohrenzerlegung entsprechen dabei den transitiven Kanten einer aufwärtsmonotonen Zeichnung von  $G$  und die Modifizierung des Spannbaums entspricht der Konstruktion einer rechtsgedrückten Einbettung [2]. Bei der Modifizierung wird die Kante zwischen Endknoten des trivialen Ohrs und dessen Elternknoten aus dem Spannbaum entfernt und den letzten Kanten zugeordnet. Danach wird eine Kante zwischen Anfangsknoten und Endknoten des trivialen Ohrs in den Spannbaum eingefügt (Abbildung 61(c)).

In einem ersten vorbereitenden Preorder-Durchlauf gegen den Uhrzeigersinn wird die Steigung bestimmt, mit der später die Kante von einem Elternknoten zu einem Kindknoten gezeichnet wird. Die Steigung wird in einem Steigungslabel des Kindknoten gespeichert. Dieser Durchlauf beginnt damit, dass der Senke die Steigung null zugeordnet wird. Ist ein Kindknoten das erste Kind seines Elternknoten, das gezeichnet wird, erhält der Kindknoten dieselbe Steigung wie sein Elternknoten; andernfalls wird die bisher höchste vergebene Steigung um eins erhöht und dem Kindknoten als Steigung zugewiesen. Hierfür wird ein globaler Zähler vorgehalten, der entsprechend aktualisiert wird. Für die Verwaltung der

letzten Kanten wird für jeden Knoten ein Stack initialisiert. Die Blätter des Spannbaums (mit Ausnahme der Senke) sind im Graphen Anfangsknoten einer letzten Kante. Gelangt man im Rahmen des Durchlaufs zu einem Blatt des Spannbaums (mit Ausnahme der Senke), wird der Endknoten der letzten Kante ermittelt, die von diesem Blatt ausgeht. Auf den Stack des so ermittelten Endknoten wird dann die entsprechende letzte Kante gelegt. Im Anfangsknoten der letzten Kante wird in einem Label—im Folgenden Kantenlabel genannt—eine Referenz auf den Endknoten geeignet gespeichert. Abbildung 62 zeigt die Belegung der Steigungs- und Kantenlabel sowie der Stacks in einem Beispiel.

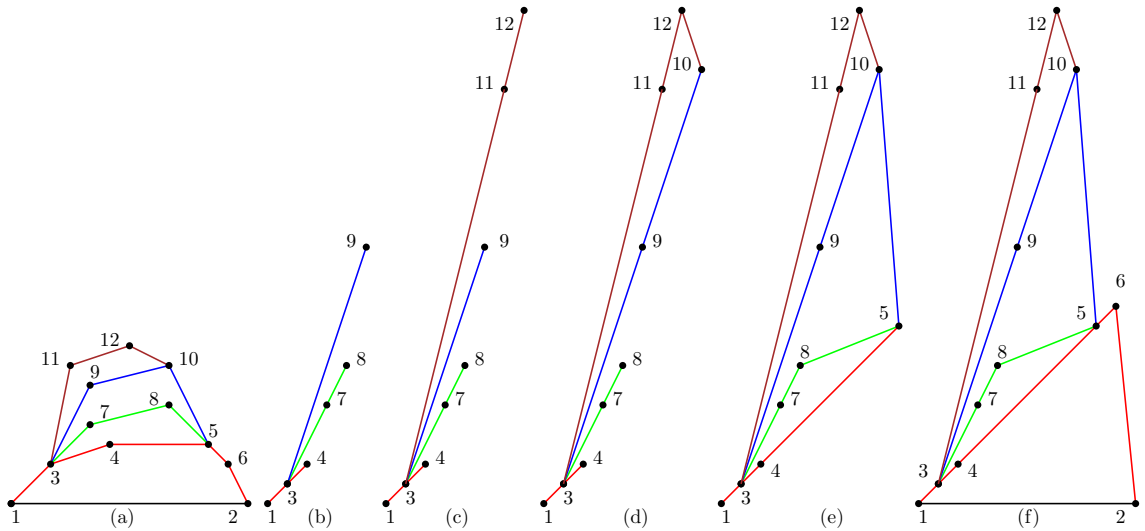


**Abbildung 62:** (a) Serien-paralleler Graph  $G$  mit verschachtelter offener Ohrenzerlegung. (b) Belegung der Labels und Stacks in der Reihenfolge Steigungslabel, Kantenlabel und Stack.

Die Zeichnung soll wie üblich so erstellt werden, dass zunächst die Kante  $(s, t)$  als waagerechte Kante von links nach rechts und dann der weitere Graph oberhalb der Kante  $(s, t)$  gezeichnet wird. Während eines zweiten Durchlaufs durch den Spannbaum werden sukzessive die Knoten, die Kanten des Spannbaumes und auch die letzten Kanten gezeichnet. Dieser zweite Durchlauf ist grundsätzlich ebenfalls als Preorder-Durchlauf gegen den Uhrzeigersinn konzipiert. Die Reihenfolge der besuchten Knoten wird jedoch im Hinblick auf die Zeichnung der letzten Kanten modifiziert. Sobald ein Knoten erreicht wird, auf dessen Stack eine oder mehrere letzte Kanten liegen, wird er zunächst nicht gezeichnet. Auch der an ihm verwurzelte Unterbaum wird nicht traversiert. Die Traversierung des Spannbaums wird vielmehr so fortgesetzt, als ob der betrachtete Knoten ein Blatt des Spannbaums sei. Grundsätzlich überprüft man nach Zeichnung eines jeden Knotens, ob in seinem Kantenlabel eine Referenz gespeichert ist und er somit Anfangsknoten einer letzten Kante ist. Ist dies der Fall, wird überprüft, ob im Stack des referenzierten Knotens die von dem gerade besuchten zum referenzierten Knoten verlaufende letzte Kante zuoberst liegt. Ein positives Ergebnis bedeutet, dass alle anderen Knoten, die Anfangsknoten einer zu dem referenzierten Knoten inzidenten letzten Kante sind, bereits besucht und gezeichnet worden sind. Der aktuelle Knoten, der referenzierte Knoten und alle auf seinem Stack liegenden letzten Kanten können in diesem Fall gezeichnet werden. Danach wird überprüft, ob seinerseits im Kantenlabel des referenzierten Knotens ein Knoten referenziert wird, der nach Prüfung seines Stacks gezeichnet werden kann. Dies kann sich als rekursiver, kaskadenförmiger Prozess fortsetzen. Nach Beendigung eines rekursiven Prozesses werden

in umgekehrter Reihenfolge die an den gezeichneten Knoten verwurzelten Unterbäume traversiert und gezeichnet. Dabei wird nach Zeichnung eines Knotens jeweils überprüft, ob erneut ein rekursiver Prozess zur Zeichnung von letzten Kanten gestartet werden kann. Die Kreisfreiheit in den Verschachtelungsbäumen der zugrundeliegenden verschachtelten offenen Ohrenzerlegung schließt das Auftreten eines Deadlocks aus, so dass der gesamte Spannbaum in der gewünschten Reihenfolge traversiert wird.

Die Lage eines Knoten ist eindeutig bestimmt durch seine  $x$ -Koordinate und die Steigung der Verbindungskante zwischen ihm und seinem Elternknoten im Spannbaum. Zu Beginn erhält die Quelle die Koordinaten  $(0,0)$ . Der Spannbaum wird nun in der vorstehend beschriebenen Weise durchlaufen. Während des Durchlaufs werden sukzessive die Knoten, die Kanten des Spannbaums und die letzten Kanten gezeichnet. Die Steigung der Verbindungskante zwischen einem zu zeichnenden Knoten und seinem Elternknoten entnimmt man seinem Steigungslabel. Als  $x$ -Koordinate weist man dem zu zeichnenden Knoten die um eins erhöhte  $x$ -Koordinate des zuvor gezeichneten Knotens zu. In Abbildung 63 wird für einen serien-parallelen Graphen eine Zeichenfolge erstellt. Damit schließt die Beschreibung des Verlaufs von Algorithmus SPFEWSEG.



**Abbildung 63:** Zeichenfolge mit reduzierter Segmentanzahl für einen serien-parallelen Graphen.

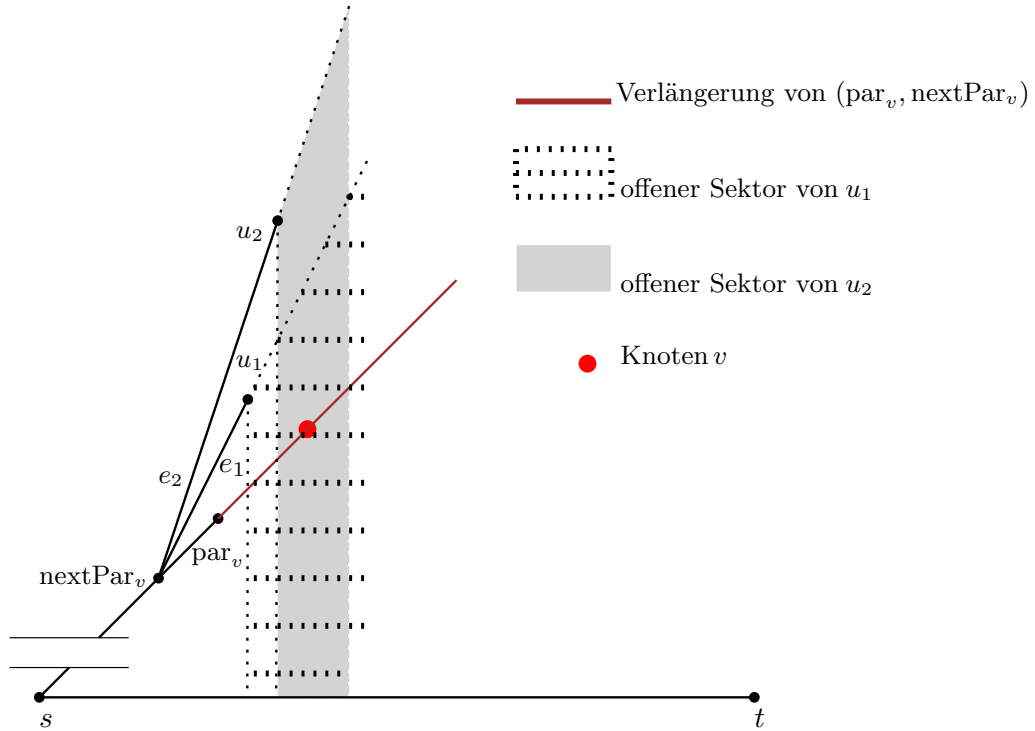
Für die weiteren Betrachtungen wird der eingeführte Begriff der ansprechenden verschachtelten offenen Ohrenzerlegung formal wie folgt definiert:

**Definition 25:** Sei  $G$  ein serien-paralleler Graph und  $(s, t)$  eine Kante gegen den Uhrzeigersinn auf dem Rand seiner Außenfläche. Unter einer von  $(s, t)$  startenden ansprechenden verschachtelten offenen Ohrenzerlegung  $\mathcal{A}$  versteht man eine verschachtelte offene Ohrenzerlegung von  $G$ , die mit  $(s, t)$  startet und folgende Eigenschaft hat:

Für  $E_{\text{par}} \in \mathcal{A}$  mit  $E_{\text{par}} \neq E_1$  sei  $E_{\text{nextPar}}$  das Ohr, in dem  $E_{\text{par}}$  verschachtelt ist. Dann verläuft jedes Ohr  $E_i$ , dass in  $E_{\text{par}}$  verschachtelt ist, außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises.

**Nachweis der Kreuzungsfreiheit der Kanten:** Als erstes werden die Kanten des Spannbaums betrachtet. Die Steigungen, mit denen die Kanten des Spannbaums gezeichnet werden, erhält man durch den vorbereitenden Preorder-Durchlauf gegen den Uhrzeigersinn. Die Vorschrift, für die Zeichnung eines neuen Asts die bisher global größte Steigung um eins zu inkrementieren, gewährleistet, dass die Kanten des Spannbaums später bei der Zeichnung untereinander keine Kreuzungen induzieren. Die im zweiten Durchlauf veränderte Besuchsreihenfolge hat darauf keinen Einfluss, da die in dem vorangegangenen Durchlauf ermittelten Steigungen nicht mehr verändert werden.

Es werden nun die letzten Kanten betrachtet. Sei ein Knoten  $v$  Endpunkt eines oder mehrerer Ohren und somit Endpunkt einer oder mehrerer letzter Kanten. Sei mit  $\text{par}_v$  sein Elternknoten im Spannbaum und mit  $\text{nextPar}_v$  dessen Elternknoten bezeichnet. Mit  $u_i, 1 \leq i \leq p$  seien die Anfangspunkte der letzten Kanten, die in  $v$  enden, notiert. Die  $u_i$  mit  $1 \leq i \leq p$  seien entsprechend ihrer Zeichenfolge nummeriert. Mit  $e_i (1 \leq i \leq p)$  sei die Kante vom Knoten  $u_i$  zu seinem Elternknoten bezeichnet. Es werde der Zeitpunkt betrachtet, an dem  $u_p$  gerade gezeichnet wurde und somit die Bedingung für die Zeichnung von  $v$  und der in  $v$  endenden letzten Kanten erfüllt ist. Dann muss für  $i < j$  mit  $i < p$  und  $j \leq p$  die  $x$ -Koordinate von  $u_j$  größer als die von  $u_i$  (Abbildung 64) sein.



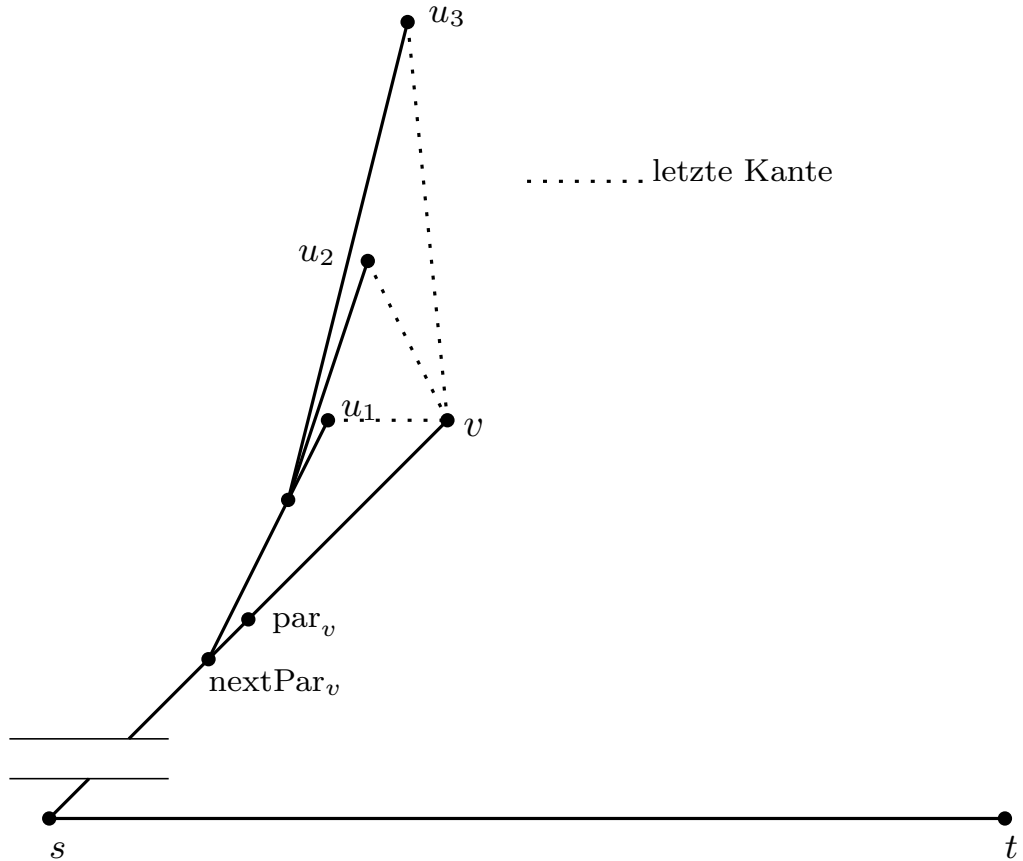
**Abbildung 64:** Offener Sektor eines Knotens.

Für alle  $1 \leq i \leq p$  liegt somit unmittelbar vor Zeichnung von  $v$  kein Knoten und keine Kante des Spannbaums in dem offenen Sektor von  $u_i$ , der durch die Verlängerung von  $e_i$  und der Senkrechten in  $u_i$  begrenzt wird. Da der Konstruktion des Spannbaums eine ansprechende verschachtelte offene Ohrenzerlegung zugrunde lag, wurde  $\text{par}_v$  in der zeitlichen Abfolge vor allen  $u_i, 1 \leq i \leq p$  gezeichnet. Die  $x$ -Koordinate von  $\text{par}_v$  ist



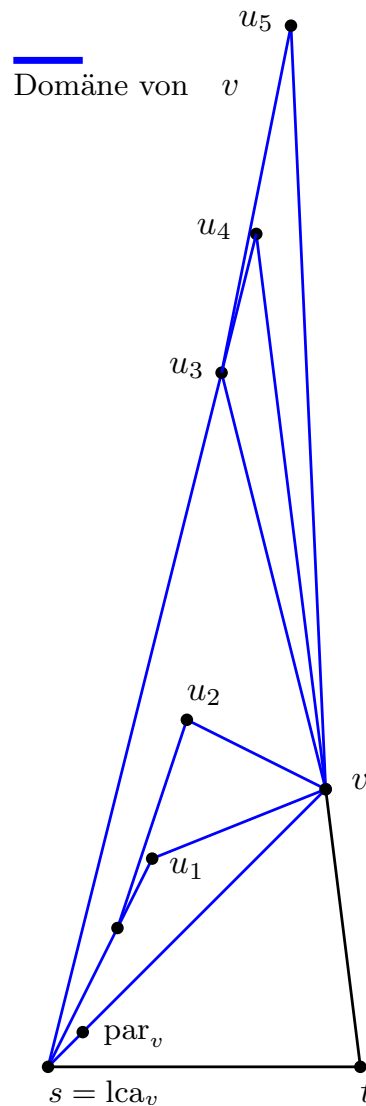
daher kleiner als die  $x$ -Koordinate von allen  $u_i, 1 \leq i \leq p$ . Somit kann der Knoten  $v$  so in Verlängerung der Kante  $(\text{nextPar}_v, \text{par}_v)$  positioniert werden, dass er in den offenen Sektoren von allen  $u_i, 1 \leq i \leq p$ , liegt. Nach Zeichnung von  $v$  induziert daher keine der zur Zeichnung anstehenden letzten Kanten eine Kreuzung mit einer Kante des Spannbaums.

Als nächstes wird nachgewiesen, dass die zu einem gemeinsamen Zielknoten inzidenten letzten Kanten sich nicht untereinander kreuzen. Seien die Bezeichnungen wie vorstehend. Es werde der Zeitpunkt betrachtet, an dem die letzte Kante  $(u_i, v)$  gerade gezeichnet wurde und die letzte Kante  $(u_{i+1}, v)$  zur Zeichnung ansteht (Abbildung 65). Der Knoten  $u_{i+1}$  hat nicht nur eine größere  $x$ -Koordinate als  $u_j$  für alle  $1 \leq j \leq i$ , sondern auch eine größere  $y$ -Koordinate. Daher kann die letzte Kante  $(u_{i+1}, v)$  so gezeichnet werden, dass sie keine der letzten Kanten  $(u_j, v)$  mit  $1 \leq j \leq i$  kreuzt. Mit Induktion folgt, dass die zu einem gemeinsamen Endpunkt inzidenten letzten Kanten untereinander keine Kreuzungen induzieren.



**Abbildung 65:** Letzte Kanten, die zu einem gemeinsamen Endpunkt inzident sind.

Es verbleibt noch der Nachweis, dass letzte Kanten mit unterschiedlichen Zielknoten sich nicht kreuzen. Seien die Bezeichnungen wieder wie vorstehend. Sei mit  $\text{lca}_i$  der niedrigste gemeinsame Vorfahre von  $u_i$  und  $v$  bezeichnet. Zunächst kann man beobachten, dass es sich bei  $\text{lca}_i$  für alle  $i$  mit  $1 \leq i \leq p$  um denselben Knoten handelt. Er wird daher im Folgenden mit  $\text{lca}_v$  bezeichnet. Man definiert nun als Domäne von  $v$  das Gebiet, das vom Weg  $(\text{lca}_v, \dots, v)$ , der Kante  $(v, u_p)$  und dem Weg  $(u_n, \dots, \text{lca}_v)$  begrenzt wird (Abbildung 66). Dabei bestehen die inneren Knoten des Weges  $(u_n, \dots, \text{lca}_v)$  aus einer Teilmenge der Knoten  $u_i, 1 \leq i < p$ . Sämtliche letzte Kanten, die  $v$  als ihren Zielknoten haben, liegen in dieser Domäne.



**Abbildung 66:** Definition der Domäne eines Knotens.

Da der Konstruktion des Spannbaums eine ansprechende verschachtelte offene Ohrenzerlegung zugrunde lag und somit der Weg  $(\text{lca}_v, \dots, \text{par}_v)$  in der zeitlichen Abfolge vor allen  $u_i$  mit  $1 \leq i \leq p$  gezeichnet wurde, liegt die Domäne vollständig oberhalb des Weges  $(\text{lca}_v, \dots, v)$ . Ferner hat  $v$  die größte  $x$ -Koordinate von allen Knoten in der Domäne von  $v$ .

Bei späterem rekursiven Aufbau von neuen Domänen kann daher die bereits bestehende Domäne eines Knotens nicht in den offenen Sektor dieses Knotens oder anderer Knoten hineinragen (Abbildung 67). Somit können die letzten Kanten einer neu zu bildenden Domäne die letzten Kanten bereits bestehender Domänen nicht kreuzen. Daher können sich letzte Kanten mit unterschiedlichen Endpunkten nicht kreuzen. Ein Nachweis der Kreuzungsfreiheit der erstellten Zeichnung ist damit erbracht.

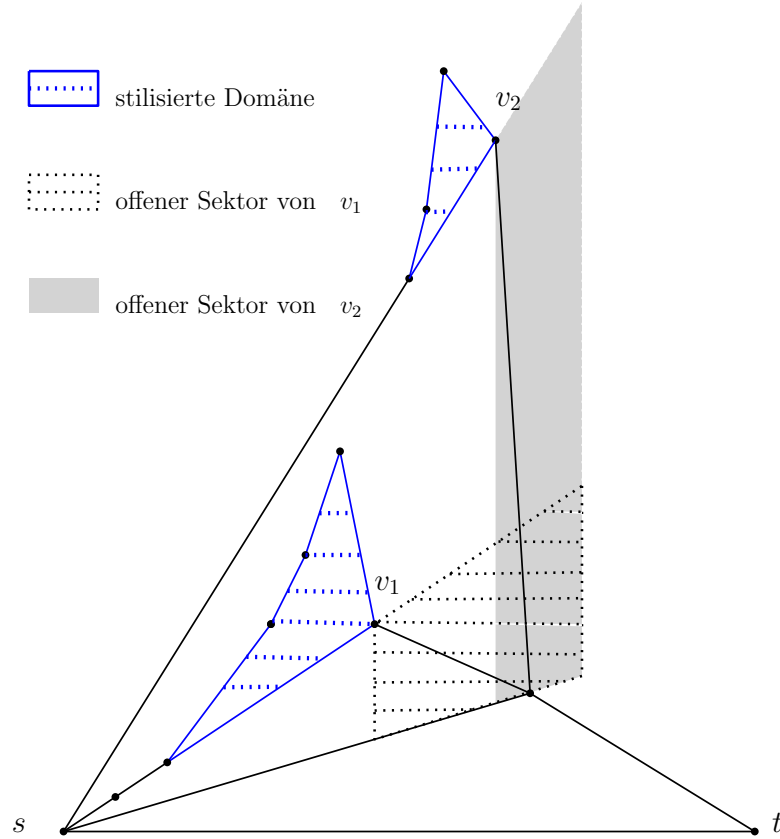
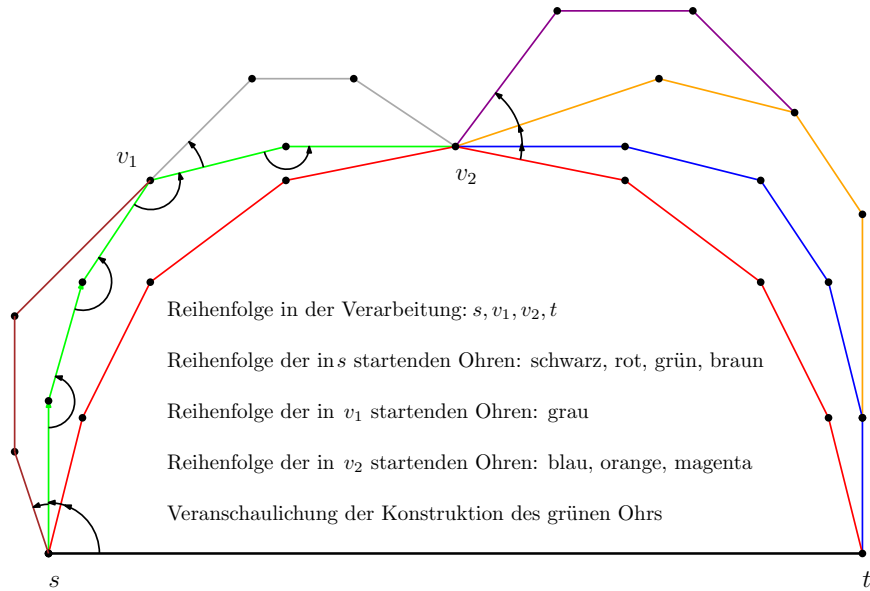


Abbildung 67: Kreuzungsfreiheit der letzten Kanten.

### 2.5.3. Ohrenzerlegung durch Algorithmus FEASIBLEDECOMPOSITION

Wie beschrieben ist eine Voraussetzung für den Ablauf von Algorithmus SPFEWSEG, dass vorab für den zu zeichnenden serien-parallelen Graphen eine ansprechende verschachtelte offene Ohrenzerlegung konstruiert wurde. Jedoch erzeugen weder Algorithmen, die mittels Breitensuche eine offene Ohrenzerlegung berechnen [29], noch Algorithmen, die hierfür Tiefensuche benutzen [5][39], für serien-parallele Graphen eine ansprechende verschachtelte offene Ohrenzerlegung. Sei zunächst wieder angenommen, dass die Kante  $(s, t)$  gegen den Uhrzeigersinn auf dem Rand der Außenfläche von  $G$  liegt. Algorithmus FEASIBLEDECOMPOSITION konstruiert für einen allgemeinen 2-zusammenhängenden Graphen  $G$  eine von  $(s, t)$  startende offene Ohrenzerlegung und für 2-zusammenhängende serien-parallele Graphen eine von  $(s, t)$  startende ansprechende verschachtelte offene Ohrenzerlegung. Darüber hinaus sind die Ohren in  $s$ - $t$ -Richtung orientiert.

**Beschreibung von Algorithmus FEASIBLEDECOMPOSITION:** Der übergeordnete Programmablauf von Algorithmus FEASIBLEDECOMPOSITION wird über einen Stack gesteuert, auf dem sich die Knoten des Graphen befinden, die aktuell verarbeitet werden können. Der Stack wird zu Beginn mit der Quelle initialisiert. Der Grundgedanke ist, bei Verarbeitung eines Knoten alle Ohren zu erzeugen, die diesen Knoten als Anfangsknoten haben. Dabei wird das jeweilige Ohr auf eine Weise erzeugt, dass später die Zerlegung für den Fall eines serien-parallelen Graphen eine ansprechende verschachtelte offene Ohrenzerlegung ist. Im Detail unterscheidet sich das Vorgehen für die Quelle einerseits und für die weiteren Knoten andererseits. Als Erstes wird die Quelle vom Stack genommen. Wie üblich ist die Kante  $(s, t)$  das erste Ohr. Daraufhin werden Quelle und Senke markiert. Die weiteren von der Quelle ausgehenden Ohren werden nun in einer Reihenfolge sukzessive konstruiert, die der Anordnung (gegen den Uhrzeigersinn) ihrer jeweils ersten Kanten um die Quelle herum entspricht. Das einzelne Ohr wird folgendermaßen erzeugt: Man nehme an, dass der Konstruktionsprozess dem betreffenden Ohr gerade die Kante  $(\text{pred}_v, v)$  hinzugefügt habe. Mit  $(v, \text{suc}_v)$  sei an  $v$  die nächste Kante gegen den Uhrzeigersinn bezeichnet. Der Knoten  $v$  wird nun markiert. Dann wird  $(v, \text{suc}_v)$  dem Ohr hinzugefügt. Dieser Prozess wird so lange fortgesetzt, bis er einen Knoten erreicht, der bereits markiert ist. Nun werden die inneren Knoten des gerade erzeugten Ohrs in umgekehrter Reihenfolge auf den Stack gelegt. Damit liegt der zweite Knoten dieses Ohrs zuoberst auf dem Stack. Nachdem alle von der Quelle ausgehenden Ohren erzeugt worden sind, wird der oberste Knoten  $v_i$  vom Stack genommen und alle von diesem Knoten ausgehenden Ohren werden erzeugt.



**Abbildung 68:** Reihenfolge während der Konstruktion der Ohren.

Dies geschieht, indem zunächst das Ohr bestimmt wird, in dem  $v_i$  innerer Knoten ist. Nun identifiziert man in diesem Ohr die von  $v_i$  ausgehende Kante  $(v_i, \text{suc}_{v_i})$ . Die an  $v_i$  nächste Kante gegen den Uhrzeigersinn stellt die erste Kante des ersten neu erzeugten von  $v_i$  ausgehenden Ohrs dar (Abbildung 68), sofern der Zielknoten dieser Kante noch

unmarkiert ist. Das Ohr selbst wird auf dieselbe Weise erzeugt wie vorstehend bei der Quelle beschrieben. Solange der gegen den Uhrzeigersinn jeweils nächste Nachbar von  $v_i$  noch unmarkiert ist, werden weitere, von  $v_i$  ausgehende Ohren erzeugt. Sobald ein markierter Nachbar von  $v_i$  erreicht ist, sind alle in  $v_i$  endenden und alle von  $v_i$  ausgehenden Ohren konstruiert. Nach dem Ende der Verarbeitung von  $v_i$  geht der Konstruktionsprozess dann weiter mit dem nächsten Knoten auf dem Stack. Insgesamt endet er, wenn keine weiteren Knoten mehr auf dem Stack sind. Da der Graph zusammenhängend ist, wird er durch das beschriebene Vorgehen vollständig durchlaufen. Damit endet die Beschreibung von Algorithmus FEASIBLEDECOMPOSITION.

**Ansprechende verschachtelte offene Ohrenzerlegung:** Der Nachweis, dass Algorithmus FEASIBLEDECOMPOSITION eine ansprechende verschachtelte offene Ohrenzerlegung konstruiert, erfolgt in mehreren Schritten. Zunächst wird folgender Hilfssatz bewiesen:

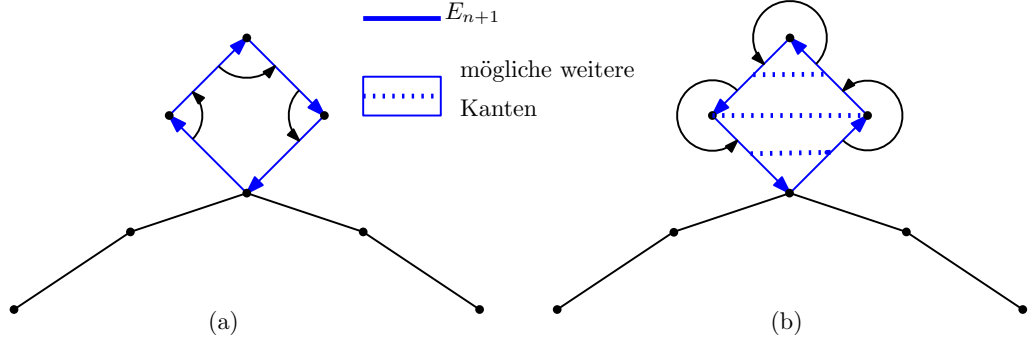
**Hilfssatz 22:** Sei  $G$  ein 2-knotenzusammenhängender Graph und  $(s, t)$  eine Kante gegen den Uhrzeigersinn auf dem Rand der Außenfläche von  $G$ . Dann konstruiert Algorithmus FEASIBLEDECOMPOSITION eine von  $(s, t)$  startende offene Ohrenzerlegung von  $G$ .

*Beweis.* Der Beweis wird induktiv geführt. Die Konstruktion des ersten Ohrs  $E_1 = (s, t)$  stellt den Basisfall dar. Offensichtlich ist die nur aus  $E_1$  bestehende Ohrenzerlegung eine offene Ohrenzerlegung.

Induktionsschluss: Man nehme nun an, dass nach der Konstruktion von  $n$  Ohren für den bis dahin traversierten Untergraphen  $G'$  von  $G$  eine offene Ohrenzerlegung  $\mathcal{D}_n$  erzeugt worden ist. Es werde nun wie beschrieben das  $(n + 1)$ -te Ohr  $E_{n+1}$  konstruiert. Mit  $\mathcal{D}_{n+1}$  sei die nun erzeugte Ohrenzerlegung bezeichnet. Ein 2-knotenzusammenhängender Graph ist auch 2-kantenzusammenhängend [39]. Da  $G$  daher auch 2-kantenzusammenhängend ist, muss der beschriebene Konstruktionsprozess für  $E_{n+1}$  nach endlich vielen Schritten einen markierten Knoten erreichen und kann nicht vorher abbrechen. Es wird nun gezeigt, dass Anfangsknoten und Endknoten des konstruierten Ohrs unterschiedlich sind. Der Beweis hierfür wird indirekt geführt. Man nehme daher an, dass Anfangs- und Endknoten des erzeugten Ohrs identisch sind. Dieser Knoten sei mit  $v_{\text{ident}}$  bezeichnet. Es werden zwei Fälle unterschieden:

- (1) Der induzierte Kreis verläuft im Uhrzeigersinn (Abbildung 69(a)). Dies führt zu einem Widerspruch, da laut Konstruktionsvorschrift bereits zuvor mit der jetzt letzten Kante des Kreises ein neues Ohr hätte beginnen müssen.
- (2) Der induzierte Kreis verläuft gegen den Uhrzeigersinn (Abbildung 69(b)). Dann können von der Kreislinie weitere Kanten nur in das Innere des Kreises hinein verlaufen, jedoch nicht heraus auf die den Kreis umgebene Fläche. Dies würde jedoch bedeuten, dass  $v_{\text{ident}}$  Artikulationsknoten von  $G$  ist, was einen Widerspruch zum 2-Zusammenhang von  $G$  darstellt.

In beiden möglichen Fällen wurde ein Widerspruch hergeleitet. Dies bedeutet, dass die Anfangs- und Endknoten von  $E_{n+1}$  nicht identisch sind. Mit der Induktionsannahme folgt, dass auch  $\mathcal{D}_{n+1}$  eine offene Ohrenzerlegung ist. Damit ist der Induktionsschluss bewiesen.



**Abbildung 69:** (a) und (b): Illustration der Beweisführung für die Korrektheit von Algorithmus FEASIBLEDECOMPOSITION.

Aus Basisfall und Induktionsschluss folgt die Behauptung. Somit konstruiert Algorithmus FEASIBLEDECOMPOSITION von einem 2-zusammenhängenden Graphen eine offene Ohrenzerlegung.  $\square$

Wenn  $G$  serien-parallel ist, gilt für die von Algorithmus FEASIBLEDECOMPOSITION erzeugte Ohrenzerlegung:

**Hilfssatz 23:** Sei  $G$  ein 2-zusammenhängender serien-paralleler Graph und  $(s, t)$  eine Kante gegen den Uhrzeigersinn auf dem Rand der Außenfläche von  $G$ . Dann konstruiert Algorithmus FEASIBLEDECOMPOSITION eine von  $(s, t)$  startende verschachtelte offene Ohrenzerlegung von  $G$ .

*Beweis.* Da  $G$  2-(knoten)zusammenhängend ist, erzeugt Algorithmus FEASIBLEDECOMPOSITION nach Hilfssatz 22 eine von  $(s, t)$  startende offene Ohrenzerlegung von  $G$ . Nach Hilfssatz 19 muss dies eine verschachtelte offene Ohrenzerlegung von  $G$  sein, was zu beweisen war.  $\square$

Im folgenden Hilfssatz 24 ist die im Hinblick auf Algorithmus SPFEWSEG wesentliche Eigenschaft von Algorithmus FEASIBLEDECOMPOSITION formuliert.

**Hilfssatz 24:** Sei  $G$  ein 2-zusammenhängender serien-paralleler Graph und  $(s, t)$  eine Kante gegen den Uhrzeigersinn auf dem Rand der Außenfläche von  $G$ . Dann konstruiert Algorithmus FEASIBLEDECOMPOSITION eine von  $(s, t)$  startende ansprechende verschachtelte offene Ohrenzerlegung von  $G$ .

*Beweis.* Nach Hilfssatz 23 erzeugt Algorithmus FEASIBLEDECOMPOSITION für  $G$  eine verschachtelte offene Ohrenzerlegung. Es muss somit noch nachgewiesen werden, dass diese Zerlegung ansprechend ist. Seien die Bezeichnungen  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  wie früher gewählt. Algorithmus FEASIBLEDECOMPOSITION hat zwei wesentliche Konstruktionsvorschriften. In der einen wird die Reihenfolge gegen den Uhrzeigersinn festgelegt, mit der die an einem Knoten beginnenden Ohren nacheinander erzeugt werden. Die andere schreibt vor, beim Voranschreiten des Konstruktionsprozesses die an einem Knoten nächste Kante gegen den Uhrzeigersinn als nächste Kante des Ohrs auszuwählen. In Abbildung 68 kann dies

insbesondere am Knoten  $v_1$  bei der Konstruktion des grün gezeichneten Ohrs beobachtet werden. Das Zusammenspiel der beiden Vorschriften gewährleistet, dass ein beliebiges in  $E_{\text{par}}$  verschachteltes Ohr außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verläuft. Die durch Algorithmus FEASIBLEDECOMPOSITION konstruierte verschachtelte offene Ohrenzerlegung von  $G$  muss also eine ansprechende verschachtelte offene Ohrenzerlegung sein, was zu beweisen war.  $\square$

Somit wurde gezeigt, dass die für den korrekten Ablauf von Algorithmus SPFEWSEG erforderliche Konstruktion einer ansprechenden verschachtelten offenen Ohrenzerlegung eines serien-parallel Graphen auch tatsächlich mit Hilfe von Algorithmus FEASIBLEDECOMPOSITION erzeugt werden kann.

**Nachweis der  $s$ - $t$ -Orientierung:** Eine Orientierung ordnet in einem ungerichteten Graphen jeder Kante eine Richtung zu. Eine  $s$ - $t$ -Orientierung ist eine Orientierung, so dass der entstehende gerichtete Graph keine Kreise hat und  $s$  bzw.  $t$  die einzige Quelle bzw. Senke sind [39]. Für die Korrektheit von Algorithmus SPFEWSEG müssen die Ohren bei der Eingabe in  $s$ - $t$ -Richtung orientiert sein. Es muss also noch nachgewiesen werden, dass die Ohren der von Algorithmus FEASIBLEDECOMPOSITION konstruierten Zerlegung in  $s$ - $t$ -Richtung orientiert sind. Wesentlich bei der Beweisführung ist die Kreisfreiheit des implizit entstehenden gerichteten Graphen.

*Beweis.* Der Beweis wird induktiv geführt. Der Basisfall—Erzeugung des Ohrs  $E_1 = (s, t)$ —ist offensichtlich wahr. Für die Induktionsannahme nehme man an, dass nach  $n$  Schritten eine ansprechende verschachtelte offene Ohrenzerlegung  $\mathcal{D}_n$  erzeugt worden ist, deren  $n$  Ohren alle in  $s$ - $t$ -Richtung orientiert sind. Werde nun das  $(n + 1)$ -te Ohr  $E_{n+1}$  erzeugt. Sei  $v_{\text{first}}$  der Anfangsknoten und  $v_{\text{last}}$  der Endknoten dieses Ohrs. Da  $G$  serien-parallel ist, gibt es ein Ohr  $E_i \in \mathcal{D}_n$ , in dem  $v_{\text{first}}$  und  $v_{\text{last}}$  Knoten sind. Es wird nun bewiesen, dass  $v_{\text{last}}$  in  $E_i$  nach  $v_{\text{first}}$  liegen muss. Zum Zweck eines Widerspruchsbeweises nehme man an, dass  $v_{\text{last}}$  in  $E_i$  vor  $v_{\text{first}}$  liegt. Dann ist  $v_{\text{last}}$  entweder die Quelle oder interner Knoten eines Ohrs, nicht notwendigerweise von  $E_i$ . Also muss  $v_{\text{last}}$  auf den Stack gelegt worden sein. Andererseits kann  $v_{\text{last}}$  noch nicht vom Stack entfernt worden sein, da  $v_{\text{last}}$  sonst nicht Endpunkt von  $E_{n+1}$  sein könnte. Also liegt  $v_{\text{last}}$  bei Erzeugung von  $E_{n+1}$  noch auf dem Stack. Bisher erzeugte Ohren können daher  $v_{\text{last}}$  nicht als Anfangsknoten haben. Somit kann  $v_{\text{last}}$  nicht Anfangsknoten von  $E_i$  sein. Daher muss  $v_{\text{last}}$  interner Knoten oder Endknoten von  $E_i$  sein. Endknoten von  $E_i$  kann  $v_{\text{last}}$  nicht sein, da  $v_{\text{last}}$  in diesem Fall in  $E_i$  nicht vor  $v_{\text{first}}$  liegen könnte. Also muss  $v_{\text{last}}$  interner Knoten von  $E_i$  sein. Man unterscheidet nun:

- (1) Der Knoten  $v_{\text{first}}$  ist Endpunkt von  $E_i$ . Dann lag  $v_{\text{first}}$  bereits auf dem Stack, als das Ohr  $E_i$  erzeugt wurde. Insbesondere lag somit  $v_{\text{first}}$  bereits auf dem Stack, als  $v_{\text{last}}$  auf den Stack gelegt wurde. Somit lag im Stack  $v_{\text{last}}$  oberhalb von  $v_{\text{first}}$ .
- (2) Der Knoten  $v_{\text{first}}$  ist interner Knoten von  $E_i$ . Dann wurden sowohl  $v_{\text{last}}$  als auch  $v_{\text{first}}$  nach Erzeugung des Ohrs  $E_i$  auf den Stack gelegt. Da  $v_{\text{last}}$  in  $E_i$  vor  $v_{\text{first}}$  liegt und laut Konstruktionsvorschrift die internen Knoten beginnend mit dem vorletzten Knoten in

umgekehrter Reihenfolge auf den Stack gelegt werden, lag im Stack auch in diesem Fall  $v_{\text{last}}$  oberhalb von  $v_{\text{first}}$ .

Jeder Knoten ist interner Knoten von höchstens einem Ohr. Er kann daher höchstens einmal auf den Stack gelegt werden. Sowohl im Fall (1) als auch im Fall (2) kann sich daher  $v_{\text{last}}$  zu dem Zeitpunkt, an dem  $v_{\text{first}}$  vom Stack genommen und das Ohr  $E_{n+1}$  erzeugt wird, nicht auf dem Stack befinden. Dies stellt einen Widerspruch dar. Es wurde also bewiesen, dass  $v_{\text{last}}$  in  $E_i$  nach  $v_{\text{first}}$  liegen muss.

Zum Zweck eines weiteren Widerspruchsbeweises nehme man nun an, dass  $E_{n+1}$  nicht in  $s$ - $t$ -Richtung orientiert sei. Dann muss dieses Ohr einen Kreis induzieren. Seien die Bezeichnungen wie vorstehend. Zuvor wurde bewiesen, dass  $v_{\text{last}}$  in  $E_i$  nach  $v_{\text{first}}$  liegt. Dann gibt es in  $E_i$  einen Weg  $P_1 = (v_{\text{first}} \rightarrow v_{\text{last}})$ . Wenn das Ohr  $E_{n+1}$  einen Kreis induziert, muss es in  $\mathcal{D}_n$  einen Weg  $P_2 = (v_{\text{last}} \rightarrow v_{\text{first}})$  geben, wobei offensichtlich  $P_1 \neq P_2$  ist. Dann aber wird durch  $P_1$  und  $P_2$  bereits in  $\mathcal{D}_n$  ein Kreis induziert. Somit waren in  $\mathcal{D}_n$  nicht alle Ohren in  $s$ - $t$ -Richtung orientiert, was einen Widerspruch zur Induktionsannahme darstellt. Also ist das Ohr  $E_{n+1}$  ebenfalls in  $s$ - $t$ -Richtung orientiert. Der Induktionsschluss ist somit bewiesen.

Aus Basisfall und Induktionsschluss folgt, dass für einen serien-parallelen Graphen die von Algorithmus FEASIBLEDECOMPOSITION erzeugten Ohren in  $s$ - $t$ -Richtung orientiert sind und daher durch Algorithmus SPFEWSEG korrekt verarbeitet werden können.  $\square$

#### 2.5.4. Folgerungen für Algorithmus SPFEWSEG

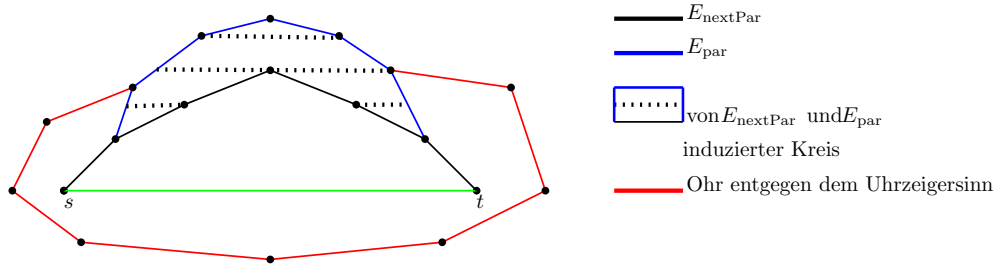
In diesem Kapitel wird aufgezeigt, wie man mit den Erkenntnissen aus dem vorangegangenen Kapitel Wahlfreiheit für die Kante  $(s, t)$  in Algorithmus SPFEWSEG erlangen und somit die bisherige Beschränkung aufheben kann, dass  $(s, t)$  gegen den Uhrzeigersinn auf dem Rand der Außenfläche liegen muss. Anschließend erfolgt die Analyse der Gitterschranke und der visuellen Komplexität in den von Algorithmus SPFEWSEG erstellten Zeichnungen.

**Freie Wahl der Bezugskante  $(s, t)$ :** Sei  $G$  ein serien-paralleler Graph und  $(s, t)$  eine beliebig gewählte Kante von  $G$ . Sei  $G'$  der zu  $G$  kombinatorisch isomorphe Graph, bei dem die Kante  $(s, t)$  sich gegen den Uhrzeigersinn auf dem Rand der Außenfläche befindet. Die Graphen  $G$  und  $G'$  unterscheiden sich dann nur durch ihre Außenfläche. Wenn man wie im Folgenden von diesem Unterschied absieht, stellt eine Zeichnung von  $G'$  daher auch eine Zeichnung von  $G$  dar. Algorithmus SPFEWSEG kann eine Zeichnung von  $G'$ —und damit auch von  $G$ —erstellen, wenn zuvor für  $G'$  eine ansprechende verschachtelte offene Ohrenzerlegung konstruiert worden ist. Hierfür soll die Adjazenzlistendarstellung von  $G$  verwendet werden. Eine daraus berechnete verschachtelte offene Ohrenzerlegung von  $G$  ist zwar auch eine verschachtelte offene Ohrenzerlegung von  $G'$ , jedoch nicht unbedingt eine ansprechende verschachtelte offene Ohrenzerlegung von  $G'$ . Daraus resultiert die Fragestellung, welche Eigenschaften eine verschachtelte offene Ohrenzerlegung von  $G$  aufweisen muss, damit sie gleichzeitig eine ansprechende verschachtelte offene Ohrenzerlegung von



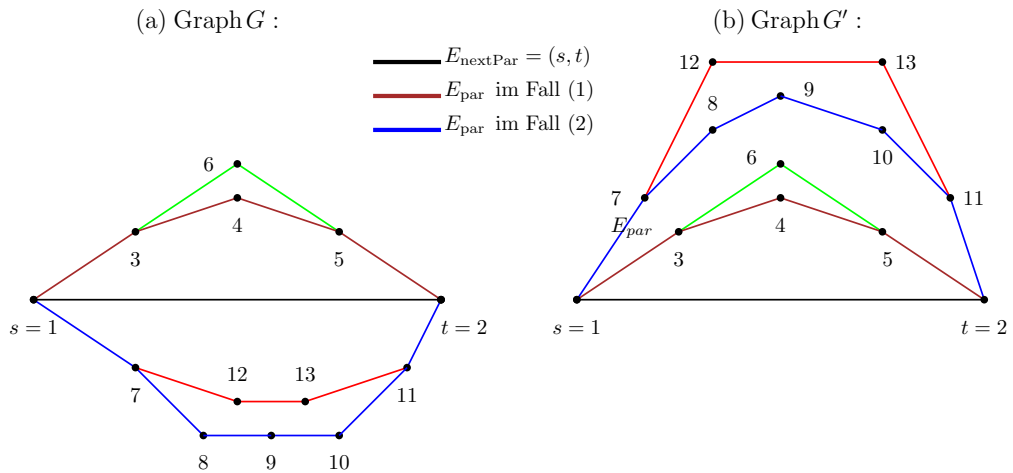
$G'$  ist. Es wird sich zeigen, dass für die Beantwortung dieser Frage die Krümmung des Ohrs berücksichtigt werden muss, in dem ein in  $G$  neu erzeugtes Ohr verschachtelt ist. Die weiteren Betrachtungen gehen implizit davon aus, dass den Ohren entsprechend ihrer  $s$ - $t$ -Orientierung eine Richtung und daraus folgend eine Krümmung gemäß Definition 23 zugeordnet wurde.

Bei der bisher geforderten Lage von  $(s, t)$  haben alle in  $E_1$  verschachtelten Ohren eine Krümmung im Uhrzeigersinn. Die weiteren durch eine ansprechende verschachtelte offene Ohrenzerlegung erzeugten Ohren verlaufen gemäß Definition 25 außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises. Da  $(s, t)$  gegen den Uhrzeigersinn auf dem Rand der Außenfläche liegt, ist es nicht möglich, dass diese Ohren ebenfalls gegen den Uhrzeigersinn gekrümmt sind. In Abbildung 70 wird dies anhand des rot gezeichneten Ohrs illustriert. Alle Ohren, die einer ansprechenden verschachtelten offenen Ohrenzerlegung angehören, haben somit eine Krümmung im Uhrzeigersinn.



**Abbildung 70:** Ein gegen den Uhrzeigersinn gekrümmtes Ohr ist nur möglich, wenn die Kante  $(s, t)$  nicht auf der Außenfläche liegt.

Sei die Kante  $(s, t)$  nun eine beliebige Kante in  $G$ . Daher wird nicht mehr gefordert, dass sie auf der Außenfläche von  $G$  liegt. Abbildung 71(a) zeigt ein solches Beispiel. In Abbildung 71(b) wird der zu  $G$  kombinatorisch isomorphe Graph  $G'$  gezeigt, in dem  $(s, t)$  gegen den Uhrzeigersinn auf dem Rand der Außenfläche liegt. Man unterscheidet nun:



**Abbildung 71:** (a) Serien-paralleler Graph  $G$ . (b) Kombinatorisch isomorpher Graph  $G'$ .

- (1) Sei  $E_{\text{par}} \neq E_1$  in einer verschachtelten offenen Ohrenzerlegung von  $G$  ein Ohr mit einer Krümmung im Uhrzeigersinn. Dieses Ohr ist in der entsprechenden verschachtelten offenen Ohrenzerlegung von  $G'$  ebenfalls im Uhrzeigersinn gekrümmt. Damit diese Zerlegung von  $G'$  eine ansprechende verschachtelte offene Ohrenzerlegung ist, muss in  $G'$  ein in  $E_{\text{par}}$  verschachteltes Ohr (in Abbildung 71 das grün gezeichnete Ohr) außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verlaufen. Dieses ist genau dann der Fall, wenn es auch in  $G$  außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verläuft.
- (2) Sei  $E_{\text{par}} \neq E_1$  in einer verschachtelten offenen Ohrenzerlegung von  $G$  ein Ohr mit einer Krümmung gegen den Uhrzeigersinn. Dieses Ohr ist in der entsprechenden verschachtelten offenen Ohrenzerlegung von  $G'$  im Uhrzeigersinn gekrümmt. Damit diese Zerlegung von  $G'$  eine ansprechende verschachtelte offene Ohrenzerlegung ist, muss in  $G'$  ein in  $E_{\text{par}}$  verschachteltes Ohr (in Abbildung 71 das rot gezeichnete Ohr) außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verlaufen. Dieses ist genau dann der Fall, wenn es in  $G$  innerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verläuft. Der Grund für die im Vergleich zum ersten Fall veränderte Lagebeziehung ist folgender: Hat in  $G$  der von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierte Kreis eine Krümmung gegen den Uhrzeigersinn, wechselt dieser Kreis beim Übergang zu  $G'$  seine Krümmung. Gleichzeitig wird das Innere dieses Kreises zu seinem Außengebiet und das Außengebiet zu seinem Inneren.

Übergeordneter Gesichtspunkt war es, die Anforderungen an eine verschachtelte offene Ohrenzerlegung von  $G$  zu beschreiben, damit diese gleichzeitig eine ansprechende verschachtelte offene Ohrenzerlegung von  $G'$  darstellt. Entsprechend den vorstehenden Überlegungen trifft man folgende Definition:

**Definition 26:** Sei  $G$  ein serien-paralleler Graph und  $(s, t)$  eine beliebige Kante von  $G$ . Unter einer von  $(s, t)$  startenden *harmonischen verschachtelten offenen Ohrenzerlegung* versteht man eine verschachtelte offene Ohrenzerlegung von  $G$ , die mit  $(s, t)$  startet und folgende Eigenschaft hat: Sei  $E_{\text{par}} \neq E_1$  und  $E_{\text{nextPar}}$  das Ohr, in dem  $E_{\text{par}}$  verschachtelt ist. Dann gilt für jedes Ohr  $E_i$ , das in  $E_{\text{par}}$  verschachtelt ist:

- (1) Ist  $E_{\text{par}}$  im Uhrzeigersinn gekrümmt, liegt  $E_i$  außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises.
- (2) Ist  $E_{\text{par}}$  gegen den Uhrzeigersinn gekrümmt, liegt  $E_i$  innerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises.

Mit dem zuvor beschriebenen Zusammenhang zwischen der Krümmung eines Ohrs in  $G$  und der Krümmung dieses Ohrs in  $G'$  folgt in Verbindung mit Abbildung 71, dass eine harmonische verschachtelte offene Ohrenzerlegung von  $G$  gleichzeitig eine ansprechende verschachtelte offene Ohrenzerlegung von  $G'$  ist, oder formaler:

**Hilfssatz 25:** Sei  $G$  ein 2-zusammenhängender serien-paralleler Graph und  $(s, t)$  eine beliebige Kante von  $G$ . Sei  $G'$  ein zu  $G$  kombinatorisch äquivalenter Graph, bei dem  $(s, t)$  gegen den Uhrzeigersinn auf dem Rand der Außenfläche liegt. Dann ist eine von  $(s, t)$  startende harmonische verschachtelte offene Ohrenzerlegung von  $G$  gleichzeitig eine von  $(s, t)$  startende ansprechende verschachtelte offene Ohrenzerlegung von  $G'$ .

*Beweis.* Sei  $\mathcal{H}$  die von  $(s, t)$  startende harmonische verschachtelte offene Ohrenzerlegung von  $G$  und  $E_k$  ein beliebiges Ohr in  $\mathcal{H}$ . Durch  $\mathcal{H}$  wird in  $G'$  eine verschachtelte offene Ohrenzerlegung  $\mathcal{A}$  induziert, für die  $(E_k \in \mathcal{H}) \Leftrightarrow (E_k \in \mathcal{A})$  gilt. Im Folgenden wird gezeigt, dass  $\mathcal{A}$  eine ansprechende verschachtelte offene Ohrenzerlegung von  $G'$  ist. Sei wie zuvor  $E_{\text{par}} \neq E_1$  und  $E_{\text{nextPar}}$  das Ohr, in dem  $E_{\text{par}}$  verschachtelt ist (Abbildung 71). Ferner sei  $E_i$  ein Ohr, das in  $E_{\text{par}}$  verschachtelt ist. Hinsichtlich der Krümmung von  $E_{\text{par}}$  in  $G$  wird folgende Fallunterscheidung getroffen :

- (1) Das Ohr  $E_{\text{par}}$  ist in  $G$  im Uhrzeigersinn gekrümmt. Wegen  $E_i \in \mathcal{H}$  liegt  $E_i$  nach Definition 26 außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises. Da  $E_{\text{par}}$  auch in  $G'$  im Uhrzeigersinn gekrümmt ist, liegt  $E_i$  auch in  $G'$  außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises.
- (2) Das Ohr  $E_{\text{par}}$  ist in  $G$  gegen den Uhrzeigersinn gekrümmt. Wegen  $E_i \in \mathcal{H}$  liegt  $E_i$  nach Definition 26 innerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises. Wie zuvor beschrieben wird beim Übergang von  $G$  zu  $G'$  das innere Gebiet dieses Kreises zu seinem äußeren Gebiet, so dass  $E_{\text{par}}$  in  $G'$  im Uhrzeigersinn gekrümmt ist und  $E_i$  in  $G'$  außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verläuft.

Da in beiden Fällen  $E_i$  außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verläuft, ist  $\mathcal{A}$  nach Definition 25 eine ansprechende verschachtelte offene Ohrenzerlegung, was zu beweisen war.  $\square$

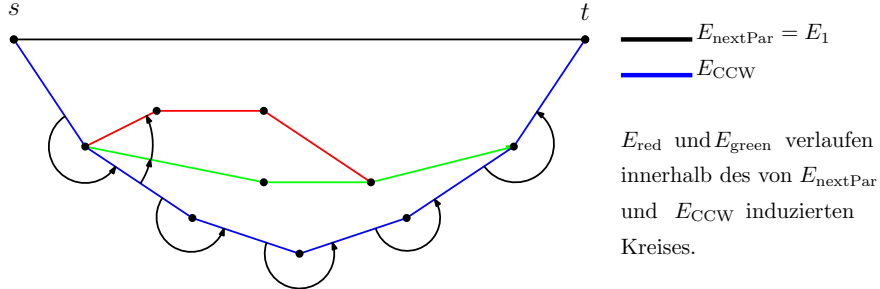
Wenn für  $G$  eine harmonische verschachtelte offene Ohrenzerlegung und somit nach Hilfssatz 25 für  $G'$  eine ansprechende verschachtelte offene Ohrenzerlegung vorliegt, kann Algorithmus SPFEWSEG ohne Weiteres eine Zeichnung von  $G'$  erstellen. Es verbleibt somit die Fragestellung, wie eine solche harmonische verschachtelte offene Ohrenzerlegung von  $G$  konstruiert werden kann. Erfreulicherweise kann hierfür Algorithmus FEASIBLEDECOMPOSITION unverändert übernommen werden. Es gilt:

**Hilfssatz 26:** Algorithmus FEASIBLEDECOMPOSITION konstruiert für einen 2-zusammenhängenden serien-parallelen Graphen  $G$  bei beliebiger Lage von  $(s, t)$  eine von  $(s, t)$  startende harmonische verschachtelte offene Ohrenzerlegung von  $G$ .

*Beweis.* Wenn  $(s, t)$  gegen den Uhrzeigersinn auf dem Rand der Außenfläche liegt, erzeugt Algorithmus FEASIBLEDECOMPOSITION für allgemeine 2-zusammenhängende Graphen nach Hilfssatz 22 eine von  $(s, t)$  startende offene Ohrenzerlegung. Die Restriktion bei der Wahl von  $(s, t)$  wurde für den Beweis von Hilfssatz 22 nicht verwendet. Der Beweis bleibt daher auch gültig, wenn  $(s, t)$  eine beliebige Lage hat. Algorithmus FEASIBLEDECOMPOSITION

konstruiert somit für allgemeine 2-zusammenhängende Graphen auch bei beliebiger Lage von  $(s, t)$  eine von  $(s, t)$  startende offene Ohrenzerlegung. Wenn  $G$  serien-parallel ist, konstruiert Algorithmus FEASIBLEDECOMPOSITION daher nach Hilfssatz 19 auch bei beliebiger Lage von  $(s, t)$  eine von  $(s, t)$  startende verschachtelte offene Ohrenzerlegung. Es verbleibt somit der Nachweis, dass es sich dabei um eine harmonische verschachtelte offene Ohrenzerlegung handelt. Seien  $E_{\text{par}} \neq E_1$  und  $E_{\text{nextPar}}$  wieder wie vorstehend definiert. Um den geforderten Nachweis zu erbringen, muss die Lage eines in  $E_{\text{par}}$  verschachtelten Ohrs bezüglich des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises untersucht werden. Entsprechend Definition 26 trifft man folgende Fallunterscheidung:

- (1) Das Ohr  $E_{\text{par}}$  ist im Uhrzeigersinn gekrümmt. In diesem Fall ist ein direkter Bezug auf den Beweis zu Hilfssatz 24 gegeben. Die dort geforderte Beschränkung auf den Fall, dass  $(s, t)$  gegen den Uhrzeigersinn auf der Außenfläche von  $G$  liegt, ist gerade gleichbedeutend mit dem Fall, dass  $E_{\text{par}}$  im Uhrzeigersinn gekrümmt ist (Abbildung 70). Dem dortigen Beweis entnimmt man, dass dann ein beliebiges in  $E_{\text{par}}$  verschachteltes Ohr außerhalb des von  $E_{\text{par}}$  und  $E_{\text{nextPar}}$  induzierten Kreises verläuft. Diese Aussage stellt gemäß Definition 26 den ersten Teil des Nachweises dar, dass die durch Algorithmus FEASIBLEDECOMPOSITION konstruierte verschachtelte offene Ohrenzerlegung von  $G$  eine harmonische verschachtelte offene Ohrenzerlegung ist.
- (2) Das Ohr  $E_{\text{par}}$  ist gegen den Uhrzeigersinn gekrümmt. Dann wird es in  $E_{\text{CCW}}$  umbenannt. Abbildung 72 veranschaulicht die Konstruktion von  $E_{\text{CCW}}$ . Während des



**Abbildung 72:** Konstruktion von  $E_{\text{CCW}}$  und der in  $E_{\text{CCW}}$  verschachtelten Ohren.

Voranschreitens bei der Konstruktion wird am aktuellen Knoten immer der folgende Nachbar gegen den Uhrzeigersinn—bezogen auf den vorangegangenen Knoten—als nächster Knoten gewählt. Im Ergebnis ist dann  $E_{\text{CCW}}$  so konstruiert, dass dann später die von einem internen Knoten von  $E_{\text{CCW}}$  ausgehenden Ohren innerhalb des von  $E_{\text{CCW}}$  und  $E_{\text{nextPar}}$  induzierten Kreises liegen. Dieser Sachverhalt stellt gemäß Definition 26 den zweiten Teil des Nachweises dar, dass die durch Algorithmus FEASIBLEDECOMPOSITION konstruierte verschachtelte offene Ohrenzerlegung von  $G$  eine harmonische verschachtelte offene Ohrenzerlegung ist.

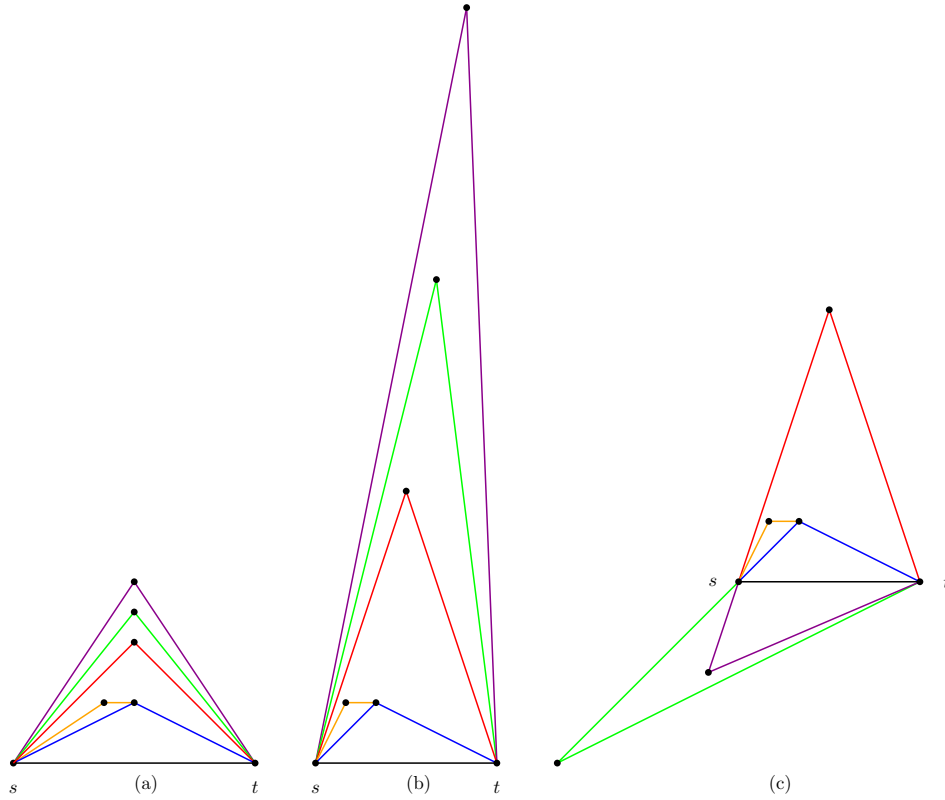
Beide Fälle zusammen beweisen, dass Algorithmus FEASIBLEDECOMPOSITION für  $G$  und beliebiger Lage von  $(s, t)$  eine von  $(s, t)$  startende harmonische verschachtelte offene Ohrenzerlegung konstruiert.  $\square$

Es bleibt also festzuhalten, dass Algorithmus FEASIBLEDECOMPOSITION von einem 2-zusammenhängenden serien-parallelen Graphen  $G$  für eine beliebige Kante  $(s, t)$  eine von  $(s, t)$  startende harmonische verschachtelte offene Ohrenzerlegung erzeugt. Nach Hilfssatz 25 erzeugt Algorithmus FEASIBLEDECOMPOSITION somit von dem zu  $G$  kombinatorisch äquivalenten Graphen  $G'$ , bei dem  $(s, t)$  gegen den Uhrzeigersinn auf dem Rand der Außenfläche liegt, eine ansprechende verschachtelte offene Ohrenzerlegung. Mit dieser Zerlegung als Eingabe kann Algorithmus SPFEWSEG ohne Weiteres eine Zeichnung von  $G'$  erstellen, bei der  $s$  die Quelle und  $t$  die Senke ist. Wenn man nicht zwischen einem Graphen und einem zu ihm kombinatorisch isomorphen Graphen differenziert, ist diese Zeichnung gleichzeitig eine Zeichnung von  $G$ , bei der  $s$  die Quelle und  $t$  die Senke ist. Die Kante  $(s, t)$  kann also bei Zeichnung von  $G$  durch Algorithmus SPFEWSEG frei gewählt werden, was zu zeigen war.

**Gittergröße:** Die Quelle wird mit den Koordinaten  $(0,0)$  gezeichnet. Da die  $x$ -Koordinate insgesamt  $(n - 1)$ -mal inkrementiert wird, beträgt die Breite der Zeichnung  $(n - 1)$ . Dies ist aufgrund der  $(s-t)$ -Orientierung der Ohren in der zugrunde liegenden Ohrenzerlegung auch die  $x$ -Koordinate der Senke. Da die Steigung einer Kante höchstens  $(n - 1)$  betragen kann, ist die Höhe der Zeichnung höchstens  $(n - 1)^2$ . In asymptotischer Notation beträgt die Gittergröße somit  $O(n) \times O(n^2)$ .

**Visuelle Komplexität:** Man setze voraus, dass keine trivialen Ohren vorhanden sind. Dann zeichnet der Algorithmus den serien-parallelen Graph so, dass das erste Ohr mit einem Segment gezeichnet wird und alle weiteren Ohren mit zwei Segmenten. Triviale Ohren ändern diesen Zusammenhang letztlich nicht, da ein triviales Ohr aufgrund der erforderlichen Umordnung des Spannbaums letztendlich doch mit zwei Kanten gezeichnet wird. Wenn jedes Ohr  $E_i \neq E_1$  außer Anfangsknoten und Endknoten nur einen einzigen weiteren Knoten aufweist, führt Algorithmus SPFEWSEG also nicht zu einer Verringerung der Segmentanzahl. Ein derartiger serien-paralleler Graph ist ein 2-Baum (Abbildung 73(a) und (b)).

Es gibt 2-Bäume, für die bei fixierter Einbettung auch im Nicht-Gitterfall grundsätzlich keine segmentreduzierte Zeichnung erstellt werden kann. Jede Zeichnung eines solchen 2-Baums hat  $(2n - 3)$  Segmente und damit soviel Segmente wie Kanten (Lemma 10 in [12]). Bei variabler Einbettung jedoch kann auch ein 2-Baum grundsätzlich segmentreduziert gezeichnet werden (Satz 11 in [12]). Abbildung 73(c) zeigt eine solche segmentreduzierte Zeichnung des 2-Baums aus Abbildung 73(a). Diese Zeichnung stellt die Übertragung des Algorithmus aus [12] (dortiger Satz 11) auf das Gitter dar. Bei komplexeren 2-Bäumen sind im Gitterfall im Verlauf dieses Algorithmus Skalierungen notwendig. Da die Anzahl der Skalierungen nicht durch ein logarithmisches Maß beschränkt ist und eine Schranke für den Skalierungsfaktor nicht angegeben werden kann, können allgemeine 2-Bäume nach jetzigem Erkenntnisstand auch bei variabler Einbettung nicht mit verminderter Segmentanzahl auf einem Gitter mit polynomialer Größe gezeichnet werden.



**Abbildung 73:** (a) Ein 2-Baum  $G$ . (b) Zeichnung von  $G$  mit Algorithmus SPFEWSEG. (c) Zeichnung von  $G$  mit verminderter Segmentanzahl.

**Zusammenfassung:** Wenn man nicht zwischen einem Graphen und einem zu ihm kombinatorisch isomorphen Graphen differenziert, kann das erzielte Ergebnis im folgenden Satz 6 zusammengefasst werden:

**Satz 6:** Algorithmus SPFEWSEG fertigt von einem 2-zusammenhängenden serien-parallelen Graphen  $G = (V, E)$  mit  $|V| = n$  und beliebiger Kante  $(s, t) \in E$  eine Zeichnung  $\Gamma$  an, die folgende Eigenschaften hat:

- (1) Eventuell vorhandene transitive Kanten verändern die Einbettung. Im Einzelfall können erhebliche Änderungen auftreten.
- (2) Die Quelle  $s$  wird mit den Koordinaten  $(0,0)$  gezeichnet, die Senke  $t$  mit den Koordinaten  $(0,n-1)$ . Alle übrigen Knoten haben eine  $x$ -Koordinate zwischen 0 und  $(n-1)$  und eine positive  $y$ -Koordinate.
- (3) Die Gittergröße beträgt  $O(n) \times O(n^2)$ .
- (4) Die Anzahl der Ohren in einer (beliebigen) verschachtelten offenen Ohrenzerlegung des Graphen werde mit  $\xi$  bezeichnet. Dann beträgt in  $\Gamma$  die Anzahl  $s$  der Segmente  $s = 2\xi - 1$ .
- (5) Ist  $G$  ein 2-Baum, wird keine Verringerung der visuellen Komplexität erreicht.

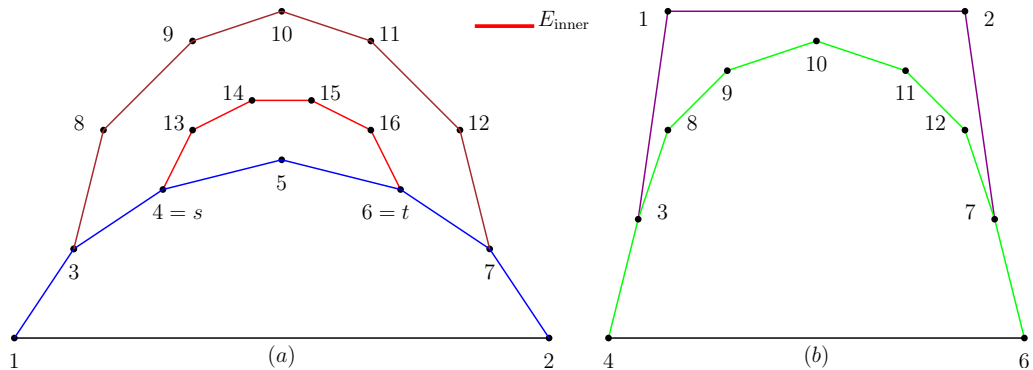
### 2.5.5. Algorithmus SPDEG3FEWSEG

Wenn der Grad des 2-zusammenhängenden serien-parallelen Graphen  $G$  gleich drei ist, lässt sich eine weitergehende Reduzierung der Segmentanzahl erreichen. Für den Nicht-Gitterfall ist dies in [38] beschrieben. Die dem Algorithmus SPDEG3FEWSEG zugrundeliegende Idee ist im Wesentlichen eine Übertragung der dort beschriebenen Idee auf den Gitterfall. In [38] wird der serien-parallele Graph mittels eines SPQ-Baums beschrieben, während für Algorithmus SPDEG3FEWSEG eine Ohrenzerlegung zur Dekomposition gewählt wird. Im konkreten Ablauf unterscheiden sich die beiden Algorithmen daher deutlich. Wahlfreiheit für die Kante  $(s, t)$  lässt sich nicht erzielen. Transitive Kanten führen zu einer Änderung der Einbettung. Es wird vorausgesetzt, dass  $G$  keine Schlingen und Multikanten hat und der Grad von  $G$  genau drei ist. Die beiden serien-parallelen Graphen mit Grad zwei—der triviale serien-parallele Graph und der  $K_3$ —werden hier nicht betrachtet.

**Beschreibung:** Zunächst werde eine von einer beliebigen Kante startende ansprechende verschachtelte offene Ohrenzerlegung  $ED_{\text{fea}}$  von  $G$  berechnet. Transitive Kanten erfordern gegebenenfalls eine Änderung der Einbettung. Die Quelle  $s$  und die Senke  $t$  in der letztendlich von Algorithmus SPDEG3FEWSEG erstellten Zeichnung müssen zwei Bedingungen genügen:

- (1) Sie müssen in  $ED_{\text{fea}}$  Anfangsknoten und Endknoten eines Ohrs  $E_{\text{inner}}$  sein, in dem keine anderen Ohren richtig verschachtelt oder enthalten sind.
- (2) Sei  $P = (s, \dots, t)$  der Weg von  $s$  nach  $t$  in dem Ohr, in dem  $E_{\text{inner}}$  verschachtelt ist. Dann darf an den inneren Knoten von  $P$  kein Ohr beginnen oder enden (Abbildung 74(a)).

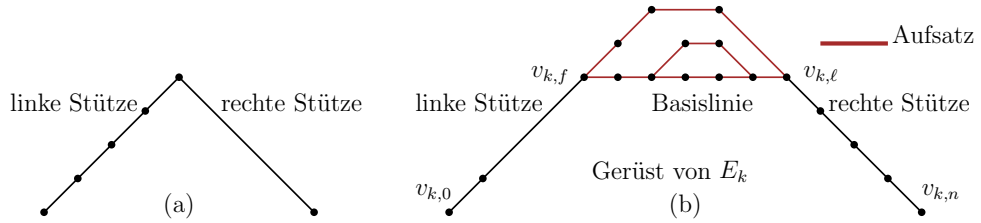
Aufgrund der hierarchischen Struktur der Verschachtelungsbäume existiert (mindestens) ein Knotenpaar, das diesen Bedingungen genügt. Sei  $s$  und  $t$  ein solches Knotenpaar. Man betrachte nun den Untergraphen  $G'$ , der aus  $G$  durch Entfernen der inneren Knoten von  $E_{\text{inner}}$  und der inneren Knoten von  $P$  entsteht. In  $G'$  existiert die Kante  $(s, t)$ . Für  $G'$  wird nun eine von  $(s, t)$  startende ansprechende verschachtelte offene Ohrenzerlegung berechnet (Abbildung 74(b)).



**Abbildung 74:** (a) Graph  $G$  mit  $E_{\text{inner}}$ . (b) Graph  $G'$  mit ansprechender verschachtelter offener Ohrenzerlegung.

Obwohl  $G'$  mit Bezug auf die ursprüngliche Quelle und Senke keine transitiven Kanten hat, kann  $G'$  mit Bezug auf  $(s, t)$  dennoch transitive Kanten haben. Dann muss die Einbettung erneut geändert werden. Die Grundidee des Algorithmus ist, zunächst eine Zeichnung von  $G'$  zu erstellen und dann die inneren Knoten von  $E_{\text{inner}}$  und  $P$  einschließlich der zu ihnen inzidenten Kanten wieder hinzuzufügen. Die Zeichnung von  $G'$  wird so erstellt, dass lediglich die Steigungen null, eins und minus eins verwendet werden. Dies ist die Grundlage für die Verringerung der Segmentanzahl. Der Algorithmus wird rekursiv in Runden ausgeführt. In jeder Runde wird ein Ohr einschließlich der in ihm richtig verschachtelten oder enthaltenen Ohren gezeichnet. Es sei vorausgesetzt, dass die Ohren in der Reihenfolge ihrer Konstruktion nummeriert wurden. Die Ohren werden dann in absteigender Nummerierung verarbeitet. Diese Reihenfolge gewährleistet, dass bei Verarbeitung eines Ohrs bereits alle Ohren verarbeitet worden sind, die in diesem Ohr richtig verschachtelt oder enthalten sind. Im einzelnen ergibt sich folgender Ablauf:

Es wird die Begriffsbildung verwendet, dass einem Ohr ein sogenanntes Gerüst zugeordnet wird. Es existieren zwei grundsätzlich unterschiedliche Arten von Gerüsten. Mittels eines Gerüsts vom Typ I werden solche Ohren gezeichnet, in denen keine weiteren Ohren verschachtelt sind. Mittels eines Gerüsts vom Typ II werden dann die übrigen Ohren einschließlich der in ihnen richtig verschachtelten oder enthaltenen Ohren gezeichnet. Die Struktur der beiden Arten von Gerüsten zeigt Abbildung 75.



**Abbildung 75:** Terminologie: (a) Gerüst vom Typ I. (b) Gerüst vom Typ II.

Bei Gerüsten vom Typ II unterscheidet man den Aufsatz und die beiden Stützen, bei Gerüsten vom Typ I fehlt der Aufsatz. Die untere Begrenzungslinie eines Aufsatzes wird als Basislinie bezeichnet. Ein Aufsatz entsteht durch das später beschriebene Zusammenfügen von Gerüsten. Die linke Stütze des Gerüsts wird mit einer Steigung von eins gezeichnet, die rechte Stütze mit einer Steigung von minus eins und die Basislinie eines gegebenenfalls vorhandenen Aufsatzes wird mit einer Steigung von null gezeichnet. Linke und rechte Stütze müssen daher gleich hoch sein. Es soll nun das einem Ohr  $E_k$  zugeordnete Gerüst initialisiert werden. Bei einem Gerüst vom Typ I geschieht dies dadurch, dass alle Knoten von  $E_k$  mit Ausnahme des letzten Knoten sukzessive als Knoten der linken Stütze gezeichnet werden. Der letzte Knoten von  $E_k$  wird als (einziger) Knoten der rechten Stütze so gezeichnet, dass rechte und linke Stütze gleich hoch sind (Abbildung 75(a)). Bei Initialisierung eines Gerüsts vom Typ II wird das Gerüst mit einem Aufsatz der Höhe null gezeichnet. Hinsichtlich der folgenden Notation sei auf Abbildung 75(a) verwiesen. Man nummeriere—beginnend mit dem ersten Knoten—in aufsteigender Reihenfolge die Knoten von  $E_k$ . Dabei werde

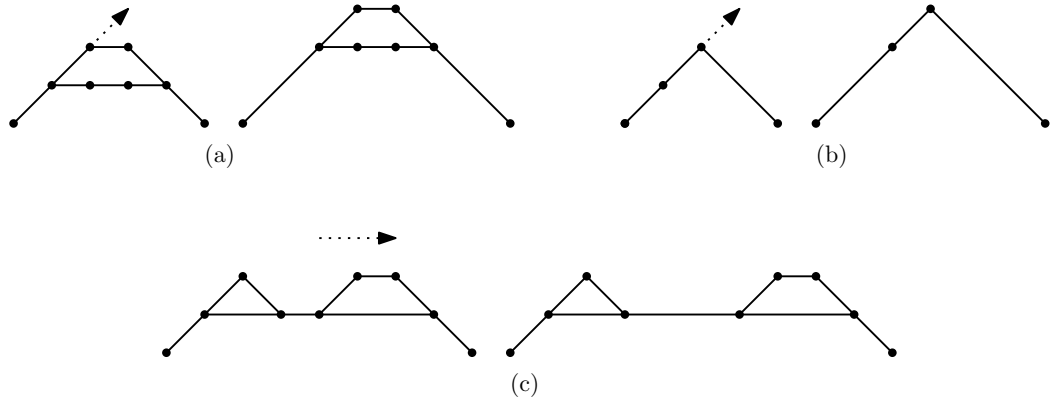


der erste Knoten mit  $v_{k,0}$ , der letzte Knoten mit  $v_{k,n}$  bezeichnet. Nun bestimme man den Knoten mit der kleinsten Nummerierung unter den Knoten, die Anfangsknoten eines in  $E_k$  richtig verschachtelten Ohrs sind. Sei dieser Knoten  $v_{k,f}$ . Ferner bestimme man den Knoten mit der größten Nummerierung unter den Knoten, die Endknoten eines in  $E_k$  richtig verschachtelten Ohrs sind. Sei dieser Knoten  $v_{k,\ell}$ . Das Maximum von  $f$  und  $(n - \ell)$  ist gleich der Höhe der beiden Stützen. Sukzessive werden nun die Knoten  $v_{k,0}$  bis  $v_{k,(f-1)}$  als Knoten der linken Stütze gezeichnet. Am Knoten  $v_{k,f}$  geht die linke Stütze in die Basislinie über. Dieser Knoten wird so gezeichnet, dass die Höhe der linken Stütze dem zuvor berechneten Wert entspricht. Weiter werden die Knoten  $v_{k,(f+1)}$  bis  $v_{k,\ell}$  aufeinanderfolgend als Knoten der Basislinie gezeichnet, bevor sukzessive die Knoten  $v_{k,(\ell+1)}$  bis  $v_{k,(n-1)}$  als Knoten der rechten Stütze gezeichnet werden. Abschließend wird der Knoten  $v_{k,n}$  wieder so als Knoten der rechten Stütze gezeichnet, dass linke und rechte Stütze gleich hoch sind. Das Gerüst von  $E_1$  hat als Sonderfall lediglich eine Basislinie und keine Stützen. Wenn im Folgenden von der Verschiebung einer dieser Strukturen gesprochen wird, soll implizit die Verschiebung der Knoten dieser Struktur gemeint sein.

Das initialisierte Gerüst des Ohrs  $E_k$  wird nun weiterverarbeitet. Hierbei wird es mit den bereits konstruierten Gerüsten der in ihm richtig verschachtelten Ohren zusammengefügt. In diesen Gerüsten korrespondiert der erste Knoten der linken Stütze und der letzte Knoten der rechten Stütze jeweils mit einem Knoten der Basislinie des Gerüsts von  $E_k$ . Insgesamt entsteht durch das Zusammenfügen eine Zeichnung von  $E_k$  einschließlich der in ihm richtig verschachtelten oder enthaltenen Ohren. Während dieses Prozesses können die Gerüste durch die folgenden Operationen verändert werden:

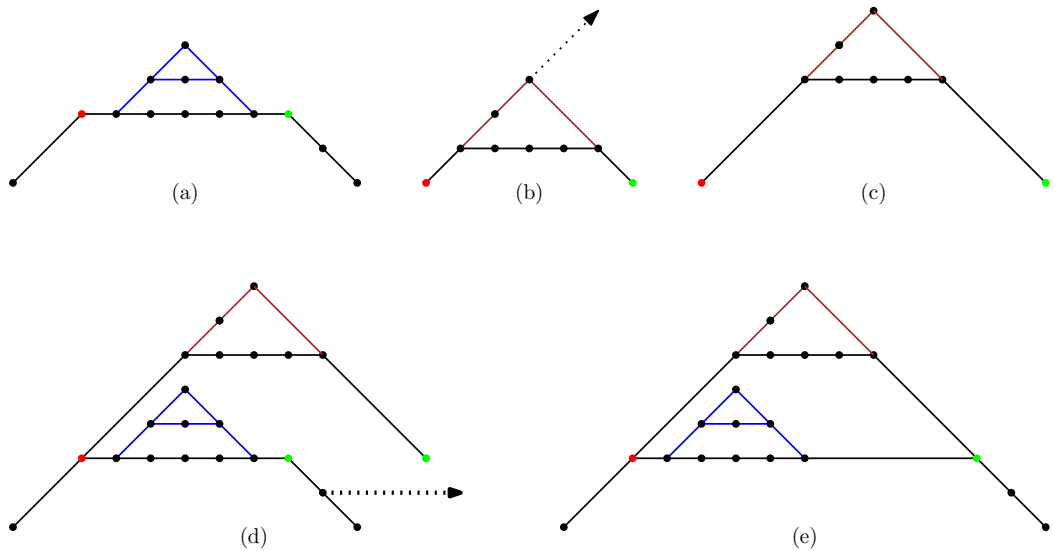
- (1) OpTranslate: Das Gerüst wird verschoben.
- (2) OpShiftSuperstructureDiagonal: Der Aufsatz wird diagonal „in die Höhe“ geschoben (Abbildung 76(a)). Linke und rechte Stütze ändern sich dementsprechend.
- (3) OpStretchTypI: Ein Gerüst vom Typ I wird in diagonalen Richtung um ein ganzzahliges Maß auseinandergezogen (Abbildung 76(b)).
- (4) OpStretchTypII: Ein Gerüst vom Typ II wird in Richtung der positiven  $x$ -Achse um ein ganzzahliges Maß auseinandergezogen. Hierfür kann ein Index in der Basislinie bestimmt werden, ab der die Knoten der Basislinie nach rechts verschoben werden. Zusammen mit diesen Knoten werden die rechte Stütze und alle Gerüste verschoben, die bereits mit dem Gerüst von  $E_k$  zusammengesetzt sind und die am spezifizierten Index oder weiter rechts beginnen (Abbildung 76(c)).

Die Reihenfolge, in der das initialisierte Gerüst von  $E_k$  mit den Gerüsten der in  $E_k$  richtig verschachtelten Ohren zusammengesetzt wird, bestimmt man durch einen Postorder-Durchlauf des Verschachtelungsbaums von  $E_k$ . Man betrachte den Zeitpunkt, an dem das Gerüst von  $E_k$  mit dem Gerüst eines (beliebigen) in  $E_k$  richtig verschachtelten Ohrs  $E_{\text{nest}}$  zusammengesetzt wird. Aufgrund der Verarbeitungsreihenfolge in einem Postorder-Durchlauf ist zu diesem Zeitpunkt das Gerüst von  $E_k$  bereits mit den Gerüsten all derjenigen Ohren zusammengesetzt worden, deren Verschachtelungsintervall in  $E_k$  ein echtes Subintervall des Verschachtelungsintervalls von  $E_{\text{nest}}$  in  $E_k$  ist. Daher kann ohne weiteres die maximale Höhe



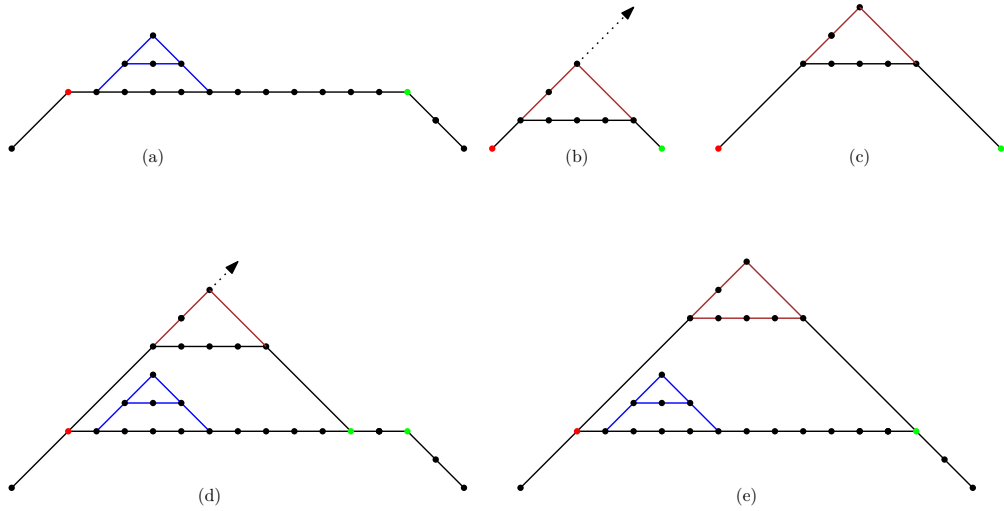
**Abbildung 76:** (a) OpShiftSuperstructureDiagonal. (b) OpStretchTypI. (c) OpStretchTypII.

der Gerüste all dieser Ohren bestimmt werden. Wenn die Höhe der Stützen des Gerüsts von  $E_{\text{nest}}$  nicht größer als diese maximale Höhe ist, wird mittels OpShiftSuperstructureDiagonal der Aufsatz des Gerüsts von  $E_{\text{nest}}$  gehoben (Abbildung 77(a)-(c)). Nun kann der erste Knoten der linken Stütze des Gerüsts von  $E_{\text{nest}}$  mit dem korrespondierenden Knoten des Gerüsts von  $E_k$  zur Übereinstimmung gebracht werden. Als nächstes soll der letzte Knoten der rechten Stütze des Gerüsts von  $E_{\text{nest}}$  mit dem korrespondierenden Knoten des Gerüsts von  $E_k$  in Übereinstimmung gebracht werden. Hierfür wird zum einen der Abstand zwischen dem ersten Knoten der linken Stütze und dem letztem Knoten der rechten Stütze im Gerüst von  $E_{\text{nest}}$  berechnet. Zum anderen wird der Abstand der beiden jeweiligen korrespondierenden Knoten in  $E_k$  bestimmt. Ist der erstgenannte Abstand größer, wird das Gerüst von  $E_k$  mittels OpStretchTypII geeignet gestreckt und so der Abstand zwischen den beiden korrespondierenden Knoten auf das richtige Maß gebracht (Abbildung 77(d)-(e)).



**Abbildung 77:** Korrespondierende Knoten sind farblich hervorgehoben. (a) Gerüst von  $E_k$ . (b) Gerüst von  $E_{\text{nest}}$ . (c) Heben des Aufsatzes des Gerüsts von  $E_{\text{nest}}$  mittels OpShiftSuperstructureDiagonal. (d)-(e) Streckung des Gerüsts von  $E_k$ .

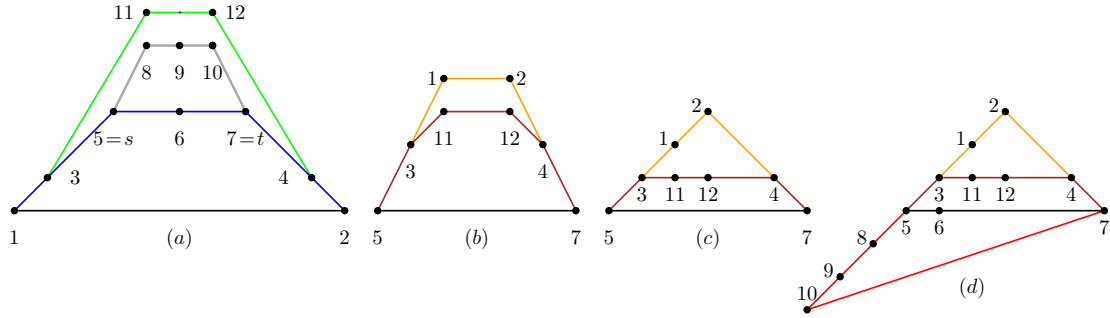
Ist der zweitgenannte Abstand um ein gerades Maß größer als der erstgenannte Abstand, wird mittels `OpShiftSuperstructureDiagonal` der Aufsatz des Gerüsts von  $E_{\text{nest}}$  um ein geeignetes Maß gehoben und so mittelbar der erste Knoten der linken Stütze und der letzte Knoten der rechten Stütze im Gerüst von  $E_{\text{nest}}$  auf den richtigen Abstand gebracht (Abbildung 78). Ist der zweitgenannte Abstand um ein ungerades Maß größer als der erstgenannte Abstand, muss zuvor das Gerüst von  $E_k$  mittels `OpStretchII` um eine Längeneinheit gestreckt werden.



**Abbildung 78:** Korrespondierende Knoten sind farblich hervorgehoben. (a) Gerüst von  $E_k$ . (b) Gerüst von  $E_{\text{nest}}$ . (c) Heben des Aufsatzes des Gerüsts von  $E_{\text{nest}}$  mittels `OpShiftSuperstructureDiagonal`. (d)-(e) Erneutes Heben des Aufsatzes des Gerüsts von  $E_{\text{nest}}$  mittels `OpShiftSuperstructureDiagonal`.

Als Ergebnis der jeweils durchgeführten Operation befindet sich der letzte Knoten der rechten Stütze des Gerüsts von  $E_{\text{nest}}$  und sein korrespondierender Knoten im Gerüst von  $E_k$  in Übereinstimmung. Damit ist das Zusammensetzen der beiden Gerüste beendet. Auf die gleiche Art wird das Gerüst von  $E_k$  mit den Gerüsten von all den Ohren zusammengesetzt, die in  $E_k$  richtig verschachtelt sind. Dieses zusammengesetzte Gebilde hat alle Konstruktionsmerkmale eines Gerüsts vom Typ II und kann seinerseits im weiteren Ablauf des Algorithmus verwendet werden. Damit schließt die Beschreibung einer Runde.

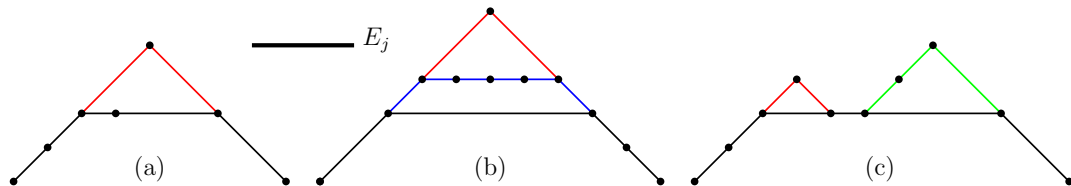
Nach Zeichnung des Gerüsts von  $E_1$  liegt eine Zeichnung des Graphen  $G'$  vor. Um daraus eine Zeichnung von  $G$  zu erhalten, muss die Zeichnung um die inneren Knoten von  $P$  und um  $E_{\text{inner}}$  ergänzt werden. Die Zeichnung der inneren Knoten kann ohne weiteres geschehen, wenn der Abstand zwischen  $s$  und  $t$  mindestens um eine Längeneinheit größer ist als die Anzahl der inneren Knoten von  $P$ . Dann können diese Knoten sukzessive auf der  $x$ -Achse—beginnend mit der Position  $(1,0)$ —gezeichnet werden. Werde nun das Ohr  $E_{\text{inner}}$  gezeichnet. Es habe insgesamt  $n$  innere Knoten. Diese werden sukzessive mit den Koordinaten  $(-i/-i)$  mit  $1 \leq i \leq n$  gezeichnet. Anschließend werden die zu diesen Knoten inzidenten Kanten gezeichnet. Die vom letzten inneren Knoten zur Senke verlaufende Kante ist die einzige Kante von  $G$ , die nicht mit einer Steigung eins, null oder minus eins gezeichnet wird (Abbildung 79).



**Abbildung 79:** (a) Graph  $G$ . (b) Graph  $G'$ . (c) Zeichnung von  $G'$ , erstellt mit Algorithmus SPDEG3FEWSEG. (d) Zeichnung von  $G$  mit Algorithmus SPDEG3FEWSEG.

Wenn dagegen der Abstand zwischen  $s$  und  $t$  nicht größer ist als die Anzahl der inneren Knoten von  $P$ , muss zuvor das dem Ohr  $E_2$  zugeordnete Gerüst gestreckt werden. Dies geschieht in Abhängigkeit von der Art dieses Gerüsts entweder mittels OpStretchTypI oder OpStretchTypII. Danach kann wie beschrieben die Zeichnung von  $G$  fertiggestellt werden. Hiermit endet die Beschreibung von Algorithmus SPDEG3FEWSEG. Es folgt die Analyse der visuellen Komplexität.

**Visuelle Komplexität:** Der triviale serien-parallele Graph und der  $K_3$  werden weiterhin nicht betrachtet. Sei  $\mathcal{B}$  die Menge der Ohren, in denen kein weiteres Ohr verschachtelt ist, und  $b$  die Anzahl dieser Ohren. Ein Ohr  $E_i \in \mathcal{B}$  ist entweder  $E_{\text{inner}}$  und wird mit einem zusätzlichen Segment gezeichnet, oder aber es wird mittels eines Gerüsts vom Typ I und daher mit zwei Segmenten gezeichnet. Alle Ohren, die Element von  $\mathcal{B}$  sind, werden daher zusammen mit  $2b - 1$  Segmenten gezeichnet. Sei nun  $\mathcal{C}$  die Menge der Ohren, in denen zumindest ein weiteres Ohr richtig verschachtelt ist, und  $c$  die Anzahl dieser Ohren. Ein Ohr  $E_j \in \mathcal{C}$  ist dann entweder  $E_1 = (s, \dots, t)$ , das mit einem Segment gezeichnet wird, oder es wird mittels eines Gerüsts vom Typ II gezeichnet. Hierbei benötigt man lediglich ein zusätzliches Segment für das Zeichnen der Basislinie. Das Zeichnen der linken bzw. rechten Stütze erfordert kein zusätzliches Segment, da sie als Verlängerung der linken bzw. rechten Stütze eines in  $E_j$  verschachtelten Ohrs gezeichnet werden. Dies ist immer möglich, da der erste und der letzte Knoten der Basislinie so gewählt wurden, dass sie Anfangsknoten bzw. Endknoten eines in  $E_j$  verschachtelten Ohrs sind. Abbildung 80 zeigt verschiedene Konstellationen.



**Abbildung 80:** Zeichnung der beiden Stützen des Gerüsts von  $E_j$  als Verlängerung von linker und rechter Stütze von ... (a) einem Gerüst des Typs I. (b) einem Gerüst des Typs II. (c) zwei Gerüsten des Typs I.

Alle Ohren, die Element von  $\mathcal{C}$  sind, werden daher zusammen mit  $c$  Segmenten gezeichnet. Die Anzahl  $d$  der Segmente in der von Algorithmus SPDEG3FEWSEG angefertigten Zeichnung von  $G$  beträgt daher  $d = c + 2b - 1$  Segmenten. Diese Anzahl soll mit [38] verglichen werden. Dort wird ein SPQ-Baum für die Zerlegung von  $G$  verwendet. Weiter wird ein primitiver P-Knoten als ein P-Knoten definiert, der im SPQ-Baum keinen P-Knoten als Nachkommen hat. Ferner ist  $P$  die Anzahl aller P-Knoten und  $N$  ist die Anzahl der primitiven P-Knoten im SPQ-Baum. Laut [38] gilt dann:

- (1) Wenn jeder S-Knoten im SPQ-Baum lediglich einen einzigen P-Knoten als direkten Nachkommen im SPQ-Baum hat, beträgt die optimale Segmentanzahl  $L = P + N + 2$ .
- (2) Andernfalls beträgt die optimale Segmentanzahl  $L = P + N + 1$ .

Die Knoten  $s$  und  $t$  wurden so gewählt, dass sie durch drei Ohren miteinander verbunden sind. Am Aufbau des P-Knoten, der die Wurzel des SPQ-Baums darstellt, sind somit drei Ohren involviert. Der Graph  $G$  hat den Grad drei. Am Aufbau jedes weiteren P-Knoten ist daher jeweils genau ein weiteres, zuvor nicht betrachtetes Ohr beteiligt. Die Anzahl  $P$  aller P-Knoten in der von Algorithmus SPDEG3FEWSEG erstellten Zeichnung ist daher  $P = c + b - 2$ . Sei nun wieder  $E_i \in \mathcal{B}$ . Man trifft folgende Fallunterscheidung:

- (1) Die Wurzel ist kein primitiver P-Knoten. Dann ist  $E_i$  entweder  $E_{\text{inner}}$  und somit nicht an einem primitiven P-Knoten beteiligt, oder es ist ein Ohr, das an genau einem primitiven P-Knoten beteiligt ist. Die Anzahl der primitiven P-Knoten beträgt somit  $N = b - 1$ . Die Anzahl  $d$  der Segmente beträgt daher in der in [38] verwendeten Notation  $d = c + 2b - 1 = (c + b - 2) + (b - 1) + 2 = P + N + 2$ .
- (2) Die Wurzel ist ein primitiver P-Knoten. Da der Grad von  $G$  genau drei ist, wird  $G$  mit vier Segmenten gezeichnet (jeweils ein Segment für  $E_1$  und  $E_{\text{inner}}$  und zwei für das dritte Ohr). Wegen  $P = N = 1$  folgt  $d = 4 = P + N + 2$ .

Für beide Fälle erhält man daher  $d = P + N + 2$ . Die Anzahl  $d$  der Segmente ist also höchstens um eins größer als die optimale Anzahl.

**Gittergröße:** Ein Gerüst wird so initialisiert, dass die Anzahl der von ihm überdeckten Spalten höchstens doppelt so groß ist wie die Anzahl seiner Knoten. Ferner wird ein Gerüst so mit einem anderen Gerüst zusammengesetzt, dass das zusammengesetzte Gebilde höchstens doppelt so viele Spalten überdeckt wie die Summe der Knoten aus beiden Gerüsten. Daraus folgt, dass die Breite der Zeichnung höchstens  $2n$  beträgt. Sieht man zunächst von  $E_{\text{inner}}$  ab, kann die Höhe der Zeichnung aufgrund der verwendeten Kantensteigungen von null, eins und minus eins ebenfalls nur höchstens  $2n$  betragen. Es ergibt sich in asymptotischer Notation eine Gittergröße von  $O(n) \times O(n)$ . Durch eine Einbeziehung von  $E_{\text{inner}}$  ändert sich diese Gitterschranke nicht.

**Zusammenfassung:** Für serien-parallele Graphen mit Grad drei konnten erheblich bessere Ergebnisse erzielt werden als für allgemeine serien-parallele Graphen. So ist die erzielte Segmentanzahl nahezu optimal, während für allgemeine serien-parallele Graphen die visuelle Komplexität schlimmstenfalls überhaupt nicht verringert wird. Auch die quadratische

Gittergröße ist im Vergleich zu der bei allgemeinen serien-parallelen Graphen erzielten kubischen Gittergröße besser. Die Ergebnisse können im folgenden Satz 7 zusammengefasst werden.

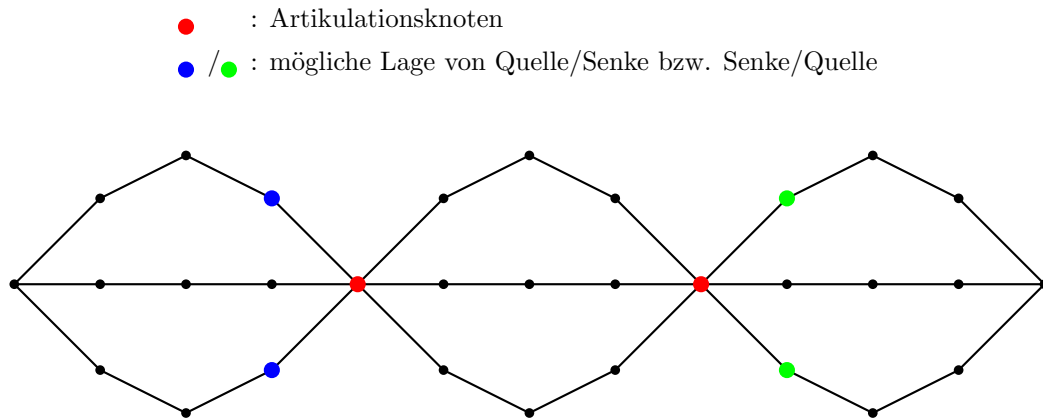
**Satz 7:** Algorithmus SPDEG3FEWSEG fertigt von einem 2-zusammenhängenden serien-parallelen Graphen  $G$  mit Grad drei eine Zeichnung mit kreuzungsfreien Kanten so an, dass die Gittergröße  $O(n) \times O(n)$  beträgt und die Segmentanzahl höchstens um eins größer ist als die optimale Segmentanzahl nach [38]. Möglicherweise wird die Einbettung dabei verändert.

### 2.5.6. Algorithmus SPFEWSEGUNIV

Im Rahmen eines universellen Algorithmus sollen nun auch serien-parallele Graphen, die nicht 2-zusammenhängend sind, mit verringerter Segmentanzahl gezeichnet werden. Gleichzeitig soll verhindert werden, dass 2-Bäume durch Algorithmus SPFEWSEG mit einer Gittergröße von  $O(n) \times O(n^2)$  gezeichnet werden, obwohl die Segmentanzahl nicht reduziert wird. Der zusammenfassende Algorithmus SPFEWSEGUNIV bestimmt zunächst die 2-zusammenhängenden Komponenten und gegebenenfalls die Artikulationsknoten eines gegebenen Graphen  $G$ . Ist dieser 2-zusammenhängend, wird eine offene Ohrenzerlegung berechnet. Mit Hilfe des Erkennungsalgorithmus aus [15] wird nun überprüft, ob es sich um eine verschachtelte offene Ohrenzerlegung und somit um einen serien-parallelen Graphen handelt. Wenn  $G$  serien-parallel ist, wird als nächstes untersucht, ob  $G$  ein 2-Baum ist. Wenn Knotenzahl  $n$  und Kantenanzahl  $m$  der Bedingung  $m = 2n - 3$  genügen und somit  $G$  ein 2-Baum ist, wird er trianguliert. Nach Berechnung eines Schnyder-Realisierers wird eine Gitterzeichnung von  $G$  auf einem  $(n - 2) \times (n - 2)$ -Gitter erstellt [40]. Dadurch wird  $G$  zwar nicht segmentreduziert gezeichnet, zumindest wird jedoch die Gittergröße minimiert. Ist  $G$  kein 2-Baum, wird sein Grad bestimmt. Ist dieser gleich drei, wird durch Algorithmus SPDEG3FEWSEG eine segmentreduzierte Zeichnung von  $G$  angefertigt, andernfalls durch Algorithmus SPFEWSEG.

Ist dagegen  $G$  nicht 2-zusammenhängend, wird im Hinblick auf Hilfssatz 20 aus seinen zuvor bestimmten 2-zusammenhängenden Komponenten und Artikulationsknoten der Baum der 2-zusammenhängenden Komponenten von  $G$  bestimmt [22]. Ist dieser Baum kein Pfad, kann  $G$  nicht serien-parallel sein. Sei dieser Baum also ein Pfad und  $G$  somit serien-parallel. Dann müssen die zu wählende Quelle  $s$  und die zu wählende Senke  $t$  den in Hilfssatz 20 formulierten Bedingungen genügen. Die Idee ist nun, dem Graphen  $G$  die zusätzliche Kante  $(s, t)$  hinzuzufügen. Der so konstruierte Graph sei mit  $G'$  bezeichnet. Für die spätere Zeichnung von  $G'$  durch Algorithmus SPFEWSEG oder SPDEG3FEWSEG berechnet Algorithmus FEASIBLEDECOMPOSITION eine ansprechende verschachtelte offene Ohrenzerlegung von  $G'$ . Implizite Voraussetzung hierfür ist, dass die Einbettung von  $G'$  eben ist. Damit die Kante  $(s, t)$  kreuzungsfrei zu  $G$  hinzugefügt werden kann und folglich die Einbettung von  $G'$  eben ist, müssen  $s$  und  $t$  noch der zusätzlichen über Hilfssatz 20

hinausgehenden Bedingung genügen, dass sie beide auf der Außenfläche von  $G$  liegen (Abbildung 81).



**Abbildung 81:** Nicht 2-zusammenhängender serien-paralleler Graph mit Artikulationsknoten und möglicher Lage von Quelle und Senke in Algorithmus SPFEWSEGUNIV.

Sei  $G'$  derart konstruiert.  $G'$  ist serien-parallel, wie folgende Überlegung zeigt.  $G$  hat nach Hilfssatz 20 eine verschachtelte offene Ohrenzerlegung  $\mathcal{D}$ , die von einem Pfad  $(s, \dots, t)$  startet. Dann ist  $\mathcal{D}' = \mathcal{D} \cup \{(s, t)\}$  eine verschachtelte offene Ohrenzerlegung von  $G'$ . Somit ist  $G'$  serien-parallel. Da  $G'$  auch 2-zusammenhängend ist, kann  $G'$  durch Algorithmus SPFEWSEGUNIV auf die gleiche Weise weiterverarbeitet werden wie zuvor  $G$  für den Fall, dass  $G$  2-zusammenhängend ist. Unter der Voraussetzung, dass  $G'$  kein 2-Baum ist, wird daher im Folgenden durch Algorithmus SPFEWSEG oder SPDEG3FEWSEG eine segmentreduzierte Zeichnung von  $G'$  angefertigt. Die segmentreduzierte Zeichnung von  $G$  erhält man letztendlich durch Entfernen der zuvor hinzugefügten Kante  $(s, t)$  aus der Zeichnung von  $G'$ . Damit schließt die Beschreibung von Algorithmus SPFEWSEGUNIV.

Im Ergebnis bleibt festzuhalten, dass Algorithmus SPFEWSEG und auch Algorithmus SPDEG3FEWSEG auf nicht 2-zusammenhängende serien-parallele Graphen übertragen werden können. Hierfür fügt man wie beschrieben die Kante  $(s, t)$  auf der Außenfläche des Graphen ein, fertigt abhängig vom Grad des Graphen die Zeichnung mit einem der beiden Algorithmen an und entfernt die eingefügte Kante wieder. Gittergröße und visuelle Komplexität ändern sich dadurch nicht.

## 3. Implementierung

Alle entwickelten Algorithmen wurden in Java 7 implementiert. Für die interaktive Eingabe wurde eine grafische Benutzeroberfläche programmiert. Als integrierte Entwicklungsumgebung wurde Eclipse in der Version Mars.1 verwendet. Die Standard-Icons wurden mit Iconion 2.6 (Free Version) mit Hilfe der dort hinterlegten Schablonen erstellt. Die anwendungsspezifischen Icons wurden mit der Open Source Software Greenfish Icon Editor Pro (Version 3.31) angefertigt.

### 3.1. Quellcode

Der Quellcode befindet sich auf der beigelegten CD im Ordner Source2523140. Als Zeichensatz wurde UTF-8 für das Projekt verwendet. Die erstellte Javadoc-Dokumentation befindet sich im Ordner Javadoc2523140. Wie üblich bietet die Datei index.html einen Überblick über die einzelnen Javadoc-Dateien.

**Paketstruktur des Quellcodes:** Die Pakete controller und view umfassen die Klassen für Steuerung und Anzeige der Applikation. Die Logik der Anwendung befindet sich in den Klassen des Pakets model. In dem Paket model.data befinden sich die verwendeten Datenstrukturen, während das Paket model.auxiliaryAlgorithms die Algorithmen für die Dekomposition und Erkennung einzelner Graphenklassen beinhaltet. Im Paket model.drawingAlgorithms befinden sich die Algorithmen für das Erstellen der segmentreduzierten Gitterzeichnungen. Die Implementierungen orientieren sich dabei eng an den Beschreibungen dieser Algorithmen in den vorangegangenen Kapiteln. Implementierungsdetails finden sich in den Kommentaren zum Quellcode und in der beiliegenden Javadoc-Dokumentation.

### 3.2. Systemvoraussetzungen und Programmstart

Das ausführbare Programm liegt als jar-Archiv mit dem Namen Diplom2523140.jar vor. Die jar-Datei benötigt zur Ausführung ein Java Environment System (JRE) mindestens der Version 7. Unter Windows muss die Systemumgebungsvariable Path angepasst sein und einen zusätzlichen Eintrag aufweisen, in dem der absolute Pfad zu dem Verzeichnis angegeben ist, in dem das JRE installiert ist. Unter Linux muss in entsprechender Weise die Datei /etc/environment editiert sein. Sowohl in Linux als auch Windows lässt sich das Programm von der Kommandozeile aus starten. Hierfür wechselt man nach Öffnen der Eingabeaufforderung in das Verzeichnis, in dem Diplom2523140.jar gespeichert ist. Durch Eingabe von `java -version` sollte zunächst überprüft werden, ob ein JRE mindestens der Version 7 installiert und die Umgebungsvariable korrekt gesetzt ist. Das Programm kann dann mit `java -jar Diplom2523140.jar` gestartet werden. Unter Windows lässt sich die jar-Datei auch ohne weiteres durch einen Doppelklick auf den entsprechenden Eintrag im Dateiverwaltungssystem starten. In Linux-Systemen—wie z.B. in Ubuntu 14.04 LTS—kann



es problematisch sein, in der Desktop-Umgebung ein geeignetes Standardprogramm für das Öffnen einer jar-Datei festzulegen. Für Linux-Systeme verbleibt es daher bei der Empfehlung, das Programm von der Kommandozeile aus zu starten.

Das Programm liegt auch als gezipptes Eclipse-Projekt mit dem Namen Eclipse2523140.zip vor. Ohne Entpacken kann es direkt in Eclipse importiert werden. Der Import kann über die Befehlsfolge *File > Import... > General > Existing Projects Into Workspace > Select archive file > Browse... > Navigieren zum Ordner Eclipse2523140.zip > Finish* erfolgen. Die Projekt-Eigenschaften werden bei dem Import automatisch konfiguriert. In Eclipse selbst muss ein JRE ebenfalls mindestens der Version 7 installiert und aktiviert sein. Die `main()`-Methode befindet sich in der Klasse `controller.Controller`. Nach Anwahl dieser Klasse kann die Anwendung dann über *Run > Run As > Java Application* gestartet werden.

### 3.3. Beschreibung des Programms

**Dateien:** Im Dateimenü sind die Operationen *Neu, Öffnen, Speichern, Speichern unter...* vorhanden. Als Dateinamenserweiterung wird das Suffix „.gra“ verwendet. Ein Dateifilter ist so konfiguriert, dass Öffnen und Speichern einer Datei nur möglich sind, wenn die Datei diese Dateinamenserweiterung hat. Codierung und Decodierung einer .gra-Datei erfolgen automatisch. Ein Graph liegt intern in einer gra-Datei im folgenden Format vor:

- (1) Die erste Zeile einer .gra-Datei besteht aus dem String „NUMBER“.
- (2) Die zweite Zeile besteht aus einer Zahl und gibt die Anzahl  $n$  der Knoten an.
- (3) Die dritte Zeile besteht aus dem String „COORDINATES“.
- (4) Die nächsten  $n$  Zeilen bestehen aus zwei Zahlen, die jeweils die  $x$ - und die  $y$ -Koordinate eines Knotens angeben. Der Zeilenblock startet mit den Koordinaten des Knotens mit der Identifikationsnummer eins und endet mit denen des Knotens mit der Nummer  $n$ .
- (5) Die nächste Zeile besteht aus dem String „EDGES“.
- (6) Jede weitere Zeile codiert eine Kante und besteht aus je zwei Zahlen. Die beiden Zahlen sind die Identifikationsnummern der beiden Endknoten der Kante. Eine (ungerichtete) Kante wird in der Datei ein zweites Mal mit vertauschten Endknoten codiert.

**Grafische Oberfläche:** Die grafische Oberfläche wird in ein Eingabefeld und ein Ausgabefeld unterteilt. Die vertikale Trennlinie kann nach rechts und links verschoben werden. Im Menü *Bearbeiten* ist eine Undo/Redo-Funktionalität implementiert. Ferner kann der Löschmodus aktiviert bzw. deaktiviert werden. Im Löschmodus bewirkt das Anklicken einer Kante oder eines Knotens im Eingabefeld das Löschen dieses Objekts. Im Menü *Ansicht* oder über die Zeichenleiste kann die Gitterdarstellung und die Nummerierung der Knoten ein- und ausgeschaltet werden. Eine Funktion zum Vergrößern bzw. Verkleinern eines Bildausschnitts ist vorhanden, ebenfalls eine Funktion zur Kantenglättung. Im Menü *Gittergröße* beziehungsweise über die Zeichenleiste kann die Auflösung des Gitternetzes variiert werden. Ein Hilfe-Dokument kann aufgerufen werden. Nachdem eine gra-Datei geöffnet oder ein Beispielsgraph über die Zeichenleiste geladen worden ist, wird der Graph

im Eingabefeld angezeigt. Alternativ kann ein neuer Graph über das Menü *Datei* angelegt werden. Die weitere Eingabe eines Graphen erfolgt grafisch. Ein Knoten wird durch Klick auf einen freien Punkt des Eingabefeldes hinzugefügt. Durch einen implementierten „Einschnapp“-Effekt wird der Knoten nicht an der Position des Mauszeigers, sondern auf dem nächsten Gitterpunkt erzeugt. Eine Kante zwischen zwei bestehenden Knoten wird eingefügt, indem man die linke Maustaste über einem bestehenden Knoten drückt, anschließend bei gedrückter linker Maustaste den Mauszeigers bis zu dem zweiten Endknoten zieht und die Maustaste über diesem Knoten loslässt. Es besteht ferner die Möglichkeit, eine neue Kante einschließlich ihres zweiten Endknotens in einer einzigen Aktion zu erzeugen. Hierfür drückt man die linke Maustaste über dem vorhandenen ersten Endknoten und zieht den Mauszeiger bei gedrückter Maustaste bis zu dem Gitterpunkt des Eingabefeldes, auf dem man den zweiten Endknoten positionieren möchte. Nach Loslassen der Maustaste erscheint im Eingabefeld der zweite Endknoten und die Verbindungskante. Für die Verschiebung der Eingabefläche verfügt das Programm über eine Autoscroll-Funktion. Damit kann während der Eingabe einer Kante das Eingabefeld verschoben werden, indem bei gedrückter linker Maustaste die Maus auf eine Begrenzung des Felds gezogen wird. Wenn sich anfangs keine Objekte außerhalb des angezeigten Bereichs befinden, ist ein Verschieben jedoch nur durch das Ziehen auf die untere oder rechte Begrenzung möglich. Durch diesen Mechanismus können Objekte auch außerhalb des aktuell angezeigten Bereichs platziert werden. Wenn sich im Eingabe- oder Ausgabefeld Objekte außerhalb des angezeigten Bereichs befinden, werden automatisch Scrollbalken eingeblendet, um zu dem Objekt navigieren zu können.

**Erstellen der Zeichnungen:** Im Eingabefeld wird der Graph mit geraden Kanten dargestellt. Während der Eingabe wird automatisch überprüft, ob ein Knoten auf oder unmittelbar neben einer bereits bestehenden Kante hinzugefügt wird. Ein entsprechender Test erfolgt nach dem Hinzufügen von Kanten im Hinblick auf Berührungen mit bereits vorhandenen Knoten. Dadurch sollen missverständliche Eingaben vermieden werden. Nach vollständiger Eingabe des Graphen kann für die betrachteten Graphenklassen eine segmentreduzierte Zeichnung berechnet und im Ausgabefeld angezeigt werden. Diese Berechnung kann durch Anwahl des entsprechenden Eintrags im Menü *Zeichnungen* oder über das diesbezügliche Icon in der Zeichenleiste gestartet werden. Es sind Menüeinträge beziehungsweise Icons für Bäume, 3-Bäume, serien-parallele Graphen, maximale außenplanare Graphen und vollständig balancierte 3-Bäume (nur im Menü) vorhanden. Mit Ausnahme der Zeichnungserstellung für Bäume setzt der weitere Programmablauf voraus, dass die Kanten des Eingabegraphen kreuzungsfrei gezeichnet sind. Ein solcher Test erfolgt nach dem Start der Zeichnungserstellung automatisch mit dem Algorithmus aus [21]. Ebenfalls automatisch wird eine Prüfung auf missverständliche Eingaben, Schlingen und Multikanten durchgeführt. Wenn Kantenkreuzungen, missverständliche Eingaben, Schlingen oder Multikanten vorhanden sind, wird der Vorgang abgebrochen und eine entsprechende Meldung ausgegeben. Die Zeichnungen selbst werden durch die in den vorangegangenen Kapiteln beschriebenen Algorithmen berechnet. Im einzelnen bedeutet dies hinsichtlich der Programmabfolge:

- (1) Nach Anwahl der Zeichnungserstellung für Bäume wird zunächst verifiziert, dass der Eingabegraph ein Baum ist. Nach erfolgreicher Verifikation erfolgt eine Aufforderung zur Wahl der Wurzel. Nach Eingabe der Wurzel wird durch Algorithmus TREEFEWSEG eine segmentreduzierte Zeichnung des Baums berechnet und im Ausgabefeld grafisch dargestellt.
- (2) Für 3-Bäume wird als erstes überprüft, ob es sich bei dem Eingabegraphen um einen 3-Baum handelt. Bei positivem Resultat fertigt Algorithmus 3TREEFEWSEG dann eine Zeichnung mit reduzierter Segmentanzahl an. Die Außenfläche wird dabei vom Eingabegraphen übernommen. Die Wurzel des Schnyder-Baums mit der geringsten Anzahl von Blättern wird auf  $(0,0)$  positioniert. Die Zerlegung in die drei Schnyder-Bäume wird farblich angezeigt.
- (3) Bei Wahl der Zeichnungserstellung für serien-parallele Graphen wird überprüft, ob der Eingabegraph serien-parallel ist. Wenn der Graph zwar serien-parallel, jedoch nicht 2-zusammenhängend ist, wird durch Algorithmus SPFEWSEGUNIV in der internen Darstellung eine Kante so hinzugefügt, dass der entstehende Graph serien-parallel und 2-zusammenhängend ist. Für den weiteren Ablauf wird folgende Fallunterscheidung getroffen:
  - (a) Wenn der serien-parallele Graph den Grad drei hat, wird ohne weitere Interaktion mit dem Benutzer durch Algorithmus SPDEG3FEWSEG eine segmentreduzierte Zeichnung erstellt. Eine Wahl der Kante  $(s, t)$  ist nicht möglich.
  - (b) Andernfalls wird der Benutzer zunächst aufgefordert, die beiden Endknoten für die Kante  $(s, t)$  zu wählen. Ist der Graph ein 2-Baum, wird er mit dem Algorithmus aus [34] trianguliert. Anschließend wird eine Gitterzeichnung mit einem Schnyder-Realisierer ohne Reduzierung der Segmentanzahl erstellt. Die gewählte Kante  $(s, t)$  liegt dabei auf der Außenfläche. In allen anderen Fällen wird durch Algorithmus SPFEWSEG eine segmentreduzierte Zeichnung angefertigt. Die gewählte Quelle hat dabei die Koordinaten  $(0,0)$ , die Senke wird auf  $(n - 1, 0)$  gezeichnet.
- (4) Für maximale außenplanare Graphen findet zunächst eine Überprüfung statt, ob der Eingabegraph tatsächlich maximal außenplanar ist. Bei positiver Verifikation wird zunächst der Hamiltonkreis des Graphen farblich hervorgehoben. Der Benutzer wird nun aufgefordert, die beiden Endpunkte der Bezugskante einzugeben. Dabei muss die Bezugskante auf dem Hamiltonkreis liegen. Nach erfolgter Eingabe wird mittels Algorithmus OUTERPLANARFEWSEG eine Zeichnung mit reduzierter Segmentanzahl erstellt und ausgegeben. Der eine Endpunkt der Bezugskante wird auf  $(0,0)$ , der andere auf  $(n - 1, 0)$  gezeichnet.
- (5) Über das Menü *Zeichnungen* können segmentminimale Zeichnungen von vollständig balancierten 3-Bäumen erstellt werden. Hierfür muss nach entsprechender Aufforderung die gewünschte Rekursionstiefe vom Benutzer eingegeben werden. Im Hinblick auf die von der Zeichnung benötigte Fläche wird die maximale Rekursionstiefe auf fünf beschränkt. Nach Eingabe der Rekursionstiefe wird die segmentminimale Zeichnung durch Algorithmus BALANCED3TREEOPTSEG berechnet und im Eingabefeld angezeigt.

## 4. Zusammenfassung und Ausblick

Eine wesentliche Fragestellung im Bereich des Graphenzeichnens ist der Zusammenhang zwischen Lesbarkeit der Zeichnung und verwirklichten Ästhetikkriterien. Von aktuellem Forschungsinteresse ist hierbei der Einfluss der visuellen Komplexität auf die Lesbarkeit. Die visuelle Komplexität einer Graphenzeichnung wird definiert als Kardinalität der Grundmenge von geometrischen Objekten, mit denen die Kanten gezeichnet werden. Ziel der vorliegenden Arbeit war es, für verschiedene Graphenklassen Algorithmen zu entwickeln, die eine Gitterzeichnung des jeweiligen Graphen mit niedriger visueller Komplexität und polynomialer Gitterschranke erzeugen. Die Kanten des Graphen sollen dabei entsprechend dem meistverbreiteten Zeichenstil als gerade Strecken gezeichnet werden. Konkretes Maß für die visuelle Komplexität solcher Zeichnungen ist dann die Anzahl der geraden Liniensegmente, mit denen die Kanten des Graphen gezeichnet werden. Die visuelle Komplexität kann dadurch verringert werden, dass mit einem einzigen geraden Liniensegment mehr als eine Kante gezeichnet wird. In einem solchen Fall hat ein Liniensegment innere Punkte. Im einzelnen wurden die folgenden kurz beschriebenen Algorithmen entworfen. Dabei sei wie üblich mit  $n$  die Anzahl der Knoten und mit  $e$  die Anzahl der Kanten bezeichnet.

- (1) Algorithmus TREEFEWSEG, der von Bäumen segmentreduzierte Zeichnungen mit einer visuellen Komplexität von  $s = \lceil 3/4 e \rceil$  und einer Gitterschranke von  $O(n^2) \times O(n^{1,58})$  erstellt. Er verwendet die Zerlegung eines Baums in schwere Pfade und das Konzept der sicheren Boxen.
- (2) Algorithmus 3TREEFEWSEG, der von 3-Bäumen segmentreduzierte Zeichnungen mit einer visuellen Komplexität von  $s = (8n - 14)/3$  bzw.  $s = 8/9 e + O(1)$  und einer Gitterschranke von  $O(n) \times O(n^2)$  erstellt. Nach Konstruktion eines Schnyder-Realisierers wird der Schnyder-Baum mit der geringsten Anzahl von Blättern so gezeichnet, dass jedes Blatt dieses Schnyder-Baums genau ein—möglicherweise mehrere Kanten umfassendes—gerades Liniensegment in der Zeichnung induziert.
- (3) Algorithmus BALANCED3TREEOPTSEG, der für vollständig balancierte 3-Bäume segmentminimale Zeichnungen mit einer Gitterschranke von  $O(n) \times O(n)$  erstellt. Während der Zeichnung des Graphen wird rekursiv ein neuer Knoten durch eine Stapeloperation im Schwerpunkt eines inneren Dreiecks positioniert.
- (4) Algorithmus OUTERPLANARFEWSEG, der für maximale außenplanare Graph Zeichnungen mit einer visuellen Komplexität von  $s = 3/2 n - 3/2$  auf einem  $O(n) \times O(n^2)$  Gitter erstellt. Der Algorithmus führt eine Zerlegung des Graphen in zwei Bäume durch. Wie bei 3-Bäumen wird dann der Baum mit der geringeren Anzahl von Blättern so gezeichnet, dass jedes Blatt genau ein gerades Liniensegment in der Zeichnung induziert.
- (5) Algorithmus SPFEWSEG, der für 2-zusammenhängende serien-parallele Graphen eine *nested open ear decomposition* des Graphen berechnet und dann eine Zeichnung erstellt, die jedes Ohr mit zwei Segmenten zeichnet. Wenn  $\xi$  die Anzahl der Ohren bezeichnet, beträgt die Gittergröße  $O(n) \times O(n^2)$  und die visuelle Komplexität  $s = 2\xi - 1$ . Der Algorithmus kann auf allgemeine serien-parallele Graphen übertragen werden.

- (6) Algorithmus SPDEG3FEWSEG, der für serien-parallele Graphen mit Grad drei nahezu segmentminimale Zeichnungen mit einer Gitterschranke von  $O(n) \times O(n)$  anfertigt. Die Kanten werden nur mit den Steigungen minus eins, null und eins gezeichnet. Durch wohlüberlegte Anordnung der zu einem Knoten inzidenten Kanten erreicht man dann die erwünschte Reduzierung der Segmentanzahl.

Der für Bäume entwickelte Algorithmus (1) ist der erste Algorithmus, der mit niedriger visueller Komplexität Zeichnungen von Bäumen auf einem polynomialen Gitter erstellt und dabei gerade Strecken für die Kanten verwendet. Die für 3-Bäume (2) und außenplanare Graphen (4) entworfenen Algorithmen sind überhaupt die ersten Algorithmen, die diese Graphen mit niedriger visueller Komplexität auf einem polynomialen Gitter zeichnen. Die drei genannten Algorithmen führen zu einer Verringerung der visuellen Komplexität, die proportional zur Anzahl der Knoten bzw. Kanten ist. Damit bewirken sie eine substantiell bessere Verringerung der visuellen Komplexität als der Algorithmus für serien-parallele Graphen (5). Bei diesem ist die Verringerung nicht proportional zur Anzahl der Knoten oder Kanten. Vielmehr ist sie abhängig von der konkreten Verteilung der Knoten auf die Ohren. Der Worst Case liegt vor, wenn jedes Ohr mit Ausnahme des ersten genau zwei Kanten hat. Dann wird die Anzahl der Liniensegmente überhaupt nicht verringert. Solche Graphen sind 2-Bäume. Im Nicht-Gitterfall gibt es 2-Bäume, die bei fixierter Einbettung überhaupt nicht segmentreduziert gezeichnet werden können. Die für 2-Bäume fehlende Verringerung der visuellen Komplexität durch Algorithmus SPFEWSEG (5) lässt sich somit gut mit den bisherigen Erkenntnissen vereinbaren. Der Algorithmus für vollständig balancierte 3-Bäume (3) und der für serien-parallele Graphen mit Grad drei (6) betreffen zwar nur eng umgrenzte Unterklassen, weisen aber eine minimale bzw. fast minimale visuelle Komplexität und eine asymptotisch optimale Gitterschranke von  $O(n) \times O(n)$  auf. Alle Algorithmen konnten prototypisch implementiert und insofern praktisch validiert werden.

Bei den Hauptklassen bleibt es Ziel, Algorithmen für das Erstellen von segmentminimalen Zeichnungen auf einem polynomialen Gitter zu entwickeln oder aber eine untere Schranke für die visuelle Komplexität anzugeben, mit der segmentreduzierte Gitterzeichnungen auf einem polynomialen Gitter angefertigt werden können. Mit dieser Intention wurde Algorithmus TREEFEWSEG zwischenzeitlich durch einen der anderen Autoren von [25] so fortentwickelt, dass segmentminimale Zeichnungen von Bäumen auf einem quasi-polynomialen Gitter erstellt werden können. Details können der genannten Referenz entnommen werden.

Alle betrachteten Graphenklassen sind planare Graphen. Es stellt sich daher die Frage, ob die Entwicklung eines entsprechenden Algorithmus auch für ebene Dreiecksgraphen möglich ist. Da ein 3-Baum auch ein ebener Dreiecksgraph ist, wäre die Anwendung von Algorithmus 3TREEFEWSEG auf ebene Dreiecksgraphen ein möglicher Ansatz. Ohne weiteres ist dies nicht möglich, da im Beweis von Algorithmus 3TREEFEWSEG mit Hilfssatz 8 eine Eigenschaft von 3-Bäumen verwendet wird, über die allgemeine ebene Dreiecksgraphen nicht verfügen. Versuchsweise wurde daher die Implementierung so abgeändert, dass Algorithmus 3TREEFEWSEG auch ebene Dreiecksgraphen als Eingabe akzeptiert. Auf diese Weise wurden einige Gitterzeichnungen von ebenen Dreiecksgraphen mit etwa 25 Knoten

erstellt. Da diese Zeichnungen korrekt waren, besteht die Vermutung, dass Algorithmus 3TREEFEWSEG auf ebene Dreiecksgraphen übertragen werden kann.

Die übergeordnete Fragestellung war der Zusammenhang zwischen Lesbarkeit einer Graphenzeichnung und niedriger visueller Komplexität. Diese Frage lässt sich letztendlich nur durch eine empirische Studie beantworten. Die in dieser Arbeit entwickelten Algorithmen können zu dem Entwurf einer solchen Studie beitragen.

## A. Inhalt der beigefügten CD

Der Inhalt der beigefügten CD ist in der folgenden Tabelle 3 aufgelistet:

**Tabelle 3:** Inhalt der beigefügten CD.

lfd. Nr.	Bezeichnung	Beschreibung
1	Diplom2523140.pdf	elektronische Version der Diplomarbeit
2	Diplom2523140.jar	ausführbare jar-Datei des Programms
3	Eclipse2523140.zip	Programm als Eclipse-Projekt
4	Source2523140	Quellcode des Programms
5	Javadoc2523140	Javadoc-Dokumentation des Programms

## Literatur

- [1] BADENT , Melanie ; BAUR , Michael ; BRANDES , Ulrik ; CORNELSEN , Sabine: Leftist Canonical Ordering. In: *Graph drawing (GD'10)*, Bd. 5849 der *Lecture Notes in Computer Science*. Springer, 2010, 159–170. [http://dx.doi.org/10.1007/978-3-642-11805-0\\_16](http://dx.doi.org/10.1007/978-3-642-11805-0_16).
- [2] BERTOLAZZI, P. ; COHEN, R. F. ; BATTISTA, G. ; TAMASSIA, R. ; TOLLIS, I. G.: How to draw a series-parallel digraph. In: *Algorithm theory - SWAT '92* Bd. 621. Springer, 1992, 272–283. [http://dx.doi.org/10.1007/3-540-55706-7\\_23](http://dx.doi.org/10.1007/3-540-55706-7_23).
- [3] BONICHON, Nicolas ; GAVOILLE, Cyril ; HANUSSE, Nicolas: Canonical Decomposition of Outerplanar Maps and Application to Enumeration, Coding, and Generation. In: *Graph-Theoretic Concepts in Computer Science* Bd. 2880. Springer, 2003, 81–92. [http://dx.doi.org/10.1007/978-3-540-39890-5\\_8](http://dx.doi.org/10.1007/978-3-540-39890-5_8).
- [4] BONICHON, Nicolas ; LE SAËC, Bertrand ; MOSBAH, Mohamed: Wagner's Theorem on Realizers. In: *Automata, Languages and Programming* Bd. 2380. Springer, 2002, 1043–1053. [http://dx.doi.org/10.1007/3-540-45465-9\\_89](http://dx.doi.org/10.1007/3-540-45465-9_89).
- [5] BRANDES, Ulrik: Eager st-Ordering. In: *Proceedings of the 10th European Symposium of Algorithms (ESA'02)* (2002), 247–256. <http://algo.uni-konstanz.de/publications/b-esto-02.pdf>
- [6] BREHM, Enno: *3-Orientations and Schnyder 3-Tree-Decompositions*, FB Mathematik und Informatik, Freie Universität Berlin, Diplomarbeit, 2000. <http://page.math.tu-berlin.de/~felsner/Diplomarbeiten/brehm.ps.gz>
- [7] CHARTRAND, Gary ; HARARY, Frank: Planar Permutation Graphs. In: *Annales de l'I.H.P. Probabilités et statistiques* 3 (1967), Nr. 4, 433–438. <http://eudml.org/doc/76875>
- [8] CHROBAK, Marek ; KANT, Goos: Convex Grid Drawings of 3-Connected Planar Graphs. In: *International Journal of Computational Geometry & Applications* 07 (1997), Nr. 03, 211–223. <http://dx.doi.org/10.1142/S0218195997000144>
- [9] COXETER, Harold S. M.: Barycentric Coordinates. § 13.7 in *Introduction to Geometry*. 2nd ed. New York: Wiley, 1969, 216–221.
- [10] DI BATTISTA, Giuseppe ; FRATI, Fabrizio: Small Area Drawings of Outerplanar Graphs. In: *Algorithmica* 54 (2009), Nr. 1, 25–53. <http://dx.doi.org/10.1007/s00453-007-9117-3>
- [11] DIESTEL, Reinhard: *Graphentheorie*. 2., neu bearb. und erw. Aufl. Berlin : Springer, 2000 (Springer-Lehrbuch).



- [12] DUJMOVIĆ, Vida ; EPPSTEIN, David ; SUDERMAN, Matthew ; WOOD, David R.: Drawings of planar graphs with few slopes and segments. In: *Computational Geometry* 38 (2007), Nr. 3, 194–212. <http://dx.doi.org/10.1016/j.comgeo.2006.09.002>
- [13] DUNCAN, Christian A. ; EPPSTEIN, David ; GOODRICH, Michael T. ; KOBOUROV, Stephen G. ; NÖLLENBURG, Martin: Drawing Trees with Perfect Angular Resolution and Polynomial Area. In: *Discrete & Computational Geometry* 49 (2013), Nr. 2, 157–182. <http://dx.doi.org/10.1007/s00454-012-9472-y>
- [14] DUROCHER, Stephane ; MONDAL, Debajyoti: Drawing Plane Triangulations with Few Segments. In: *Proceedings of the 26th Canadian Conference on Computational Geometry, CCCG 2014, Halifax, Nova Scotia, Canada* (2014), 40–45. <http://www.cccg.ca/proceedings/2014/papers/paper06.pdf>
- [15] EPPSTEIN, David: Parallel recognition of series-parallel graphs. In: *Information and Computation* 98 (1992), Nr. 1, 41–55. [http://dx.doi.org/10.1016/0890-5401\(92\)90041-D](http://dx.doi.org/10.1016/0890-5401(92)90041-D)
- [16] FÁRY, István: On straight line representation of planar graphs. In: *Acta Univ. Szeged. Sect. Sci. Math.* 11 (1948), 229–233.
- [17] FELSNER, Stefan ; TROTTER, William T.: Posets and planar graphs. In: *Journal of Graph Theory* 49 (2005), Nr. 4, 273–284. <http://dx.doi.org/10.1002/jgt.20081>
- [18] FRATI, Fabrizio: Straight-line drawings of outerplanar graphs in  $O(dn \log n)$  area. In: *Computational Geometry* 45 (2012), Nr. 9, 524–533. <http://dx.doi.org/10.1016/j.comgeo.2010.03.007>
- [19] FRAYSSEIX, H. de ; OSSONA DE MENDEZ, P.: On topological aspects of orientations. In: *Discrete Mathematics* 229 (2001), Nr. 1-3, 57–72. [http://dx.doi.org/10.1016/S0012-365X\(00\)00201-6](http://dx.doi.org/10.1016/S0012-365X(00)00201-6)
- [20] FRAYSSEIX, H. de ; PACH, J. ; POLLACK, R.: How to draw a planar graph on a grid. In: *Combinatorica* 10 (1990), Nr. 1, 41–51. <http://dx.doi.org/10.1007/BF02122694>
- [21] GATTORNA, Giacomo ; ÜNGÖR, Alper: Rensselaer Polytechnic Institute, Troy, NY, 1996. *Implementation of Sweepline Segment Intersection Algorithm in Geolab Geometrical Workbench*. [https://www.cise.ufl.edu/~ungor/papers/RPI\\_CG\\_prj.ps](https://www.cise.ufl.edu/~ungor/papers/RPI_CG_prj.ps)
- [22] HARARY, F. ; PRINS, G.: The block-cutpoint-tree of a graph. In: *Publ. Math. Debrecen* 13 (1966), 103–107.
- [23] HOBBS, Robert C. ; LOMBARDI, Mark: *Mark Lombardi - global networks*. Independent Curators International, 2004.

- [24] HOSSAIN, Md. I. ; RAHMAN, Md. S.: Straight-Line Monotone Grid Drawings of Series-Parallel Graphs. In: *Computing and combinatorics* Bd. 7936. Springer, 2013, 672–679. [http://dx.doi.org/10.1007/978-3-642-38768-5\\_59](http://dx.doi.org/10.1007/978-3-642-38768-5_59).
- [25] HÜLTENSCHMIDT, Gregor ; KINDERMANN, Philipp ; MEULEMANS, Wouter ; SCHULZ, André: Drawing Trees and Triangulations with Few Geometric Primitives. In: *Proceedings of the 32nd European Workshop on Computational Geometry (EuroCG'16)*. 2016, 55–58. [http://www.eurocg2016.usi.ch/sites/default/files/paper\\_49.pdf](http://www.eurocg2016.usi.ch/sites/default/files/paper_49.pdf).
- [26] IGAMBERDIEV, Alexander ; MEULEMANS, Wouter ; SCHULZ, André: Drawing Planar Cubic 3-Connected Graphs with Few Segments: Algorithms and Experiments. In: *Graph Drawing and Network Visualization (GD'15)*, Bd. 9411 der *Lecture Notes in Computer Science*. Springer, 2015, 113–124. [http://dx.doi.org/10.1007/978-3-319-27261-0\\_10](http://dx.doi.org/10.1007/978-3-319-27261-0_10).
- [27] KANT, G.: Drawing planar graphs using the canonical ordering. In: *Algorithmica* 16 (1996), Nr. 1, 4–32. <http://dx.doi.org/10.1007/BF02086606>
- [28] KINDERMANN, Philipp ; SCHULZ, André: *Persönliche Mitteilung*. 2016
- [29] MAON, Yael ; SCHIEBER, Baruch ; VISHKIN, Uzi: Parallel ear decomposition search (EDS) and st-numbering in graphs. In: *Theoretical Computer Science* 47 (1986), 277–298. [http://dx.doi.org/10.1016/0304-3975\(86\)90153-2](http://dx.doi.org/10.1016/0304-3975(86)90153-2)
- [30] MITCHELL, S. ; BEYER, T. ; JONES, W.: Linear Algorithms for Isomorphism of Maximal Outerplanar Graphs. In: *Journal of the ACM* 26 (1979), Nr. 4, 603–610. <http://dx.doi.org/10.1145/322154.322155>
- [31] MITCHELL, Sandra L.: Linear algorithms to recognize outerplanar and maximal outerplanar graphs. In: *Information Processing Letters* 9 (1979), Nr. 5, 229–232. [http://dx.doi.org/10.1016/0020-0190\(79\)90075-9](http://dx.doi.org/10.1016/0020-0190(79)90075-9)
- [32] MONDAL, Debajyoti ; NISHAT, Rahnuma I. ; BISWAS, Sudip ; RAHMAN, Md. S.: Minimum-segment convex drawings of 3-connected cubic plane graphs. In: *Journal of Combinatorial Optimization* 25 (2013), Nr. 3, 460–480. <http://dx.doi.org/10.1007/s10878-011-9390-6>
- [33] MONDAL, Debajyoti ; NISHAT, Rahnuma I. ; RAHMAN, Md. S. ; ALAM, Muhammad J.: Minimum-Area Drawings of Plane 3-Trees. In: *Journal of Graph Algorithms and Applications* 15 (2011), Nr. 2, 177–204. <http://dx.doi.org/10.7155/jgaa.00222>
- [34] PAJOR, Thomas: *Triangulierung eines planaren Graphen*. Universität Karlsruhe (TH), Fakultät für Informatik, 2007. [https://i11www.iti.uni-karlsruhe.de/\\_media/teaching/winter2006/algorithmengineering/triangulierung.pdf](https://i11www.iti.uni-karlsruhe.de/_media/teaching/winter2006/algorithmengineering/triangulierung.pdf). –

- [35] PATIL, H. P.: On the structure of  $k$ -trees. In: *Journal of Combinatorics Information and System Science* 11 (1986), Nr. 2-4, 57–64. <http://www.ams.org/mathscinet-getitem?mr=966069>
- [36] PURCHASE, Helen: Which aesthetic has the greatest effect on human understanding? In: *Graph Drawing (GD'97)*, Bd. 1353 der *Lecture Notes in Computer Science*. Springer, 1997, 248–261. [http://dx.doi.org/10.1007/3-540-63938-1\\_67](http://dx.doi.org/10.1007/3-540-63938-1_67).
- [37] ROSE, Donald J.: On simple characterizations of  $k$ -trees. In: *Discrete Mathematics* 7 (1974), Nr. 3-4, 317–322. [http://dx.doi.org/10.1016/0012-365X\(74\)90042-9](http://dx.doi.org/10.1016/0012-365X(74)90042-9)
- [38] SAMEE, Md. Abul H. ; ALAM, Md. J. ; ADNAN, Muhammad A. ; RAHMAN, Md. S.: Minimum Segment Drawings of Series-Parallel Graphs with the Maximum Degree Three. In: *Graph drawing GD 2015* Bd. 5417 der *Lecture Notes in Computer Science*. Springer, 2009, 408–419. [http://dx.doi.org/10.1007/978-3-642-00219-9\\_40](http://dx.doi.org/10.1007/978-3-642-00219-9_40).
- [39] SCHMIDT, Jens M.: A Simple Test on 2-vertex- and 2-edge-connectivity. In: *Inf. Process. Lett.* 113 (2013), Nr. 7, 241–244. <http://dx.doi.org/10.1016/j.ipl.2013.01.016>
- [40] SCHNYDER, Walter: Embedding planar graphs on the grid. In: *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms* (SODA 1990), 138–148. <http://dl.acm.org/citation.cfm?id=320191>
- [41] SCHULZ, André: Drawing Graphs with Few Arcs. In: *Journal of Graph Algorithms and Applications* 19 (2015), Nr. 1, 393–412. <http://dx.doi.org/10.7155/jgaa.00366>
- [42] TARJAN, Robert E.: Linking and Cutting Trees. In: *Data structures and network algorithms* Bd. 44. Philadelphia : Society for Industrial and Applied Mathematics (SIAM), 1983, 59–70. <http://dx.doi.org/10.1137/1.9781611970265.ch5>.
- [43] TUTTE, W. T.: How to Draw a Graph. In: *Proceedings of the London Mathematical Society* s3-13 (1963), Nr. 1, 743–767. <http://dx.doi.org/10.1112/plms/s3-13.1.743>
- [44] WAGNER, K.: Bemerkungen zum Vierfarbenproblem. In: *Jahresbericht der Deutschen Mathematiker-Vereinigung* 46 (1936), 26-32. <http://eudml.org/doc/146109>
- [45] ZHANG, Huaming ; HE, Xin: Canonical Ordering Trees and Their Applications in Graph Drawing. In: *Discrete & Computational Geometry* 33 (2005), Nr. 2, 321–344. <http://dx.doi.org/10.1007/s00454-004-1154-y>
- [46] ZICKFELD, Florian: *Geometric and Combinatorial Structures on Graphs*, Technische Universität Berlin, Diss., 2008. <http://dx.doi.org/10.14279/depositonce-1750>

# Erklärung

Name: Dr. Gregor Hültenschmidt

Matrikel-Nr: 2523140

Fach: Informatik

Modul: Diplomarbeit (Diplom II)

Ich erkläre, dass ich die vorliegende Diplomarbeit mit dem Thema

## **Graphzeichnen mit niedriger visueller Komplexität auf dem Gitter**

selbstständig und ohne unzulässige Inanspruchnahme Dritter verfasst habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich, inhaltlich oder sinngemäß entnommenen Stellen als solche den wissenschaftlichen Anforderungen entsprechend kenntlich gemacht. Die Versicherung selbstständiger Arbeit gilt auch für Zeichnungen, Skizzen oder graphischen Darstellungen. Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder derselben noch einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Mit der Abgabe der elektronischen Fassung der endgültigen Version der Arbeit nehme ich zur Kenntnis, dass diese mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate überprüft und ausschließlich für Prüfungszwecke gespeichert wird.

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_