



---

**Dissertation**  
**Zeichnen von Bäumen auf Gittern**

---

Marco Matzeder

25. Juni 2012



Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften an  
der Fakultät für Informatik und Mathematik der Universität Passau.

1. Gutachter: Prof. Dr. Franz J. Brandenburg, Universität Passau
2. Gutachter: PD Dr. Christian Bachmaier, Universität Passau

## Zusammenfassung

Das Zeichnen von Graphen beschäftigt sich mit der Frage, wie die durch einen Graphen repräsentierten Informationen für einen Betrachter übersichtlich und verständlich dargestellt werden können. Die Graphklasse der Bäume dient insbesondere zur Repräsentation von hierarchischen Strukturen. Neben den hierarchisch und radial darstellenden Verfahren werden Bäume auch auf dem orthogonalen Gitter gezeichnet, in welchem die Knoten auf ganzzahligen Koordinaten liegen und die Kanten entlang der horizontalen und vertikalen Gitterlinien verlaufen. Gewünscht wird eine gute Lesbarkeit der Zeichnungen und deren effiziente Berechnung. Für die formale Bewertung der Lesbarkeit existieren speziell für das Zeichnen von Bäumen definierte Ästhetikkriterien, wie eine ebene Darstellung, die Ordnungserhaltung und Kriterien zur Darstellung von Subgraphisomorphismen und Symmetrien.

Die vorliegende Arbeit befasst sich mit einer bislang wenig studierten Erweiterung des orthogonalen Gitters auf das hexagonale und oktagonale Gitter durch das Hinzunehmen von einer bzw. beider diagonalen Gitterrichtungen, und der Problemstellung, wie Bäume darauf gezeichnet werden. Dadurch können auch Bäume mit einem höheren Grad gezeichnet werden als auf dem orthogonalen Gitter. Die Einschränkung, dass nur Bäume gezeichnet werden können, deren Grad kleiner ist als die Anzahl der Gitterrichtungen des verwendeten Gitters, besteht jedoch weiterhin. Als Ästhetikkriterien werden die lokale Uniformität, die die Länge der ausgehenden Kanten eines Knotens festlegt, und Pattern, die deren Richtungen festlegen, eingeführt. Gegenüber dem bekannten linearen Flächenverbrauch von geradlinigen Zeichnungen von vollständigen Binärbäumen auf dem orthogonalen Gitter, werden für Zeichnungen von vollständigen  $d$ -ären Bäumen mit  $d > 2$  nicht-lineare untere Schranken für die benötigte Fläche auf dem hexagonalen und dem oktagonalen Gitter gezeigt. Insgesamt werden für vollständige und beliebige, geordnete und ungeordnete Bäume obere und untere Flächenschranken für Zeichnungen auf dem hexagonalen und oktagonalen Gitter präsentiert. Dabei zeigt sich, dass bei nicht-ordnungserhaltenden Zeichnungen zwar mehr als lineare, aber deutlich weniger als quadratische Fläche benötigt wird. Im Gegensatz dazu gibt es geordnete Bäume, deren ordnungserhaltende Zeichnungen exponentielle Fläche benötigen. Des Weiteren wird die Ermittlung der minimalen Zeichenfläche für geordnete  $d$ -äre Bäume ebenso als  $\mathcal{NP}$ -vollständig bewiesen, wie das Zeichnen von ungeordneten  $d$ -ären Bäumen mit einheitlichen Kantenlängen. Schließlich werden zwei Linearzeitalgorithmen vorgestellt, die geordnete  $d$ -äre Bäume unter Einhaltung der genannten Ästhetikkriterien zeichnen.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>i</b>
<b>1 Motivation</b>	<b>1</b>
<b>2 Einführung</b>	<b>5</b>
2.1 Zeichnung eines Graphen . . . . .	6
2.2 Qualität von Zeichnungen . . . . .	8
2.2.1 Zeichenkonventionen . . . . .	8
2.2.2 Ästhetikkriterien . . . . .	11
2.2.3 Nebenbedingungen . . . . .	14
2.3 Zeichenansätze verschiedener Graphklassen . . . . .	15
2.3.1 Allgemeine Graphen . . . . .	15
2.3.2 Gerichtete (azyklische) Graphen . . . . .	16
2.3.3 Planare Graphen . . . . .	18
<b>3 Zeichnen von Bäumen</b>	<b>21</b>
3.1 Bisherige Ansätze . . . . .	23
3.1.1 Hierarchisch . . . . .	23
3.1.2 Radial . . . . .	32
3.1.3 Gitter . . . . .	35
3.1.3.1 Geradlinig und orthogonal . . . . .	36
3.1.3.2 Orthogonal . . . . .	38
3.1.3.3 Geradlinig . . . . .	39
3.1.3.4 Polyline . . . . .	39
3.2 Neue Ansätze . . . . .	40
3.2.1 Vollständige Bäume . . . . .	44
3.2.2 Geordnete Bäume . . . . .	57
3.2.2.1 Flächenschranken . . . . .	57
3.2.2.2 Minimale Fläche . . . . .	61
3.2.3 Ungeordnete Bäume . . . . .	73
3.2.3.1 Fläche . . . . .	74
3.2.3.2 Einheitliche Kantenlänge . . . . .	78
3.2.3.3 Minimale Fläche . . . . .	92

<b>4</b>	<b>Zeichenverfahren auf <math>k</math>-Gitter</b>	<b>95</b>
4.1	Zeichnungen von $d$ -nären Bäumen mit Unterbaum Separation . . . . .	95
4.1.1	Definitionen . . . . .	97
4.1.2	Beschreibung des Verfahrens . . . . .	104
4.1.3	Zeitkomplexität . . . . .	108
4.1.4	Mögliche Erweiterungen . . . . .	110
4.2	Zeichnungen von $d$ -nären Bäumen ohne Unterbaum Separation . . . . .	111
4.2.1	Definitionen . . . . .	111
4.2.2	Beschreibung des Verfahrens . . . . .	113
4.2.2.1	Verkürzen der Kanten . . . . .	115
4.2.2.2	Vereinigen der Konturen . . . . .	118
4.2.3	Zeitkomplexität . . . . .	119
4.3	Zeichnungen von $d$ -nären Bäumen ohne Unterbaum Separation in Li- nearzeit . . . . .	120
4.3.1	Definitionen . . . . .	120
4.3.2	Beschreibung des Verfahrens . . . . .	121
4.3.2.1	Verkürzen der Kanten . . . . .	122
4.3.2.2	Vereinigen von Konturen . . . . .	122
4.3.2.3	Bestimmen des Schnitt- oder Projektionspunktes . . . . .	129
4.3.3	Zeitkomplexität . . . . .	129
<b>5</b>	<b>Schluss</b>	<b>131</b>
5.1	Zusammenfassung . . . . .	131
5.2	Ausblick . . . . .	132
	<b>Abbildungsverzeichnis</b>	<b>135</b>
	<b>Tabellenverzeichnis</b>	<b>139</b>
	<b>Definitionsverzeichnis</b>	<b>141</b>
	<b>Algorithmenverzeichnis</b>	<b>143</b>
	<b>Literaturverzeichnis</b>	<b>145</b>

# 1

## Motivation

---

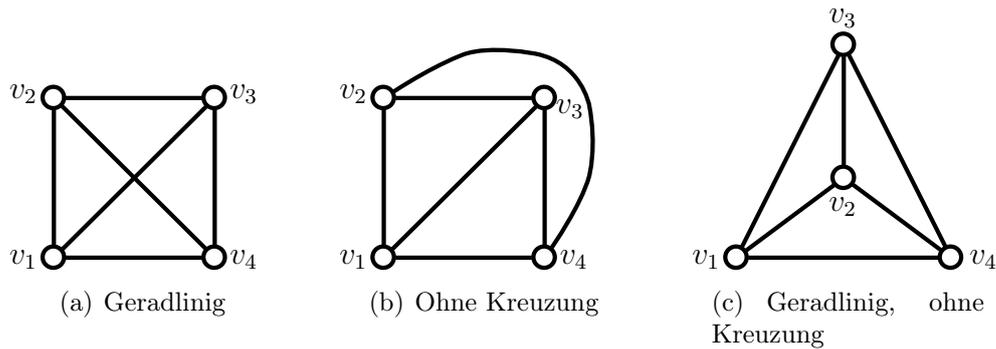
Aufgrund der Informationsflut der heutigen Zeit ist es unerlässlich Informationen und deren Zusammenhänge übersichtlich darzustellen. Daten ergeben sich in vielen Lebensbereichen vom Alltag bis hin zur Wissensrepräsentation in Informationssystemen. Sie sollen lesbar, übersichtlich und verständlich präsentiert werden, damit für den Betrachter die Hauptinformationen schnell ersichtlich werden. Die zu visualisierenden Informationen sind z. B. gespeicherte Daten, Pläne (beispielsweise U-Bahn, Bahn, Bus, Flugverkehr) oder auch Zustandsübergänge in der Biologie und Chemie. Als Beispiel für letzteres Gebiet wären beispielsweise der Zitratzyklus oder auch der Aufbau von Proteinen anzuführen. Weitere Anwendungen betreffen Webauftritte, deren Aufbau in Site Maps wiedergegeben ist oder Verzeichnisbäume, welche die Ordnerstruktur des Dateisystems präsentieren. Des Weiteren werden Diagramme in der Software Entwicklung und auch der Wissensrepräsentation verwendet. Diese werden im Speziellen zum Darstellen von Geschäftsprozessen und in der Anforderungsanalyse angefertigt, um Zusammenhänge und Abhängigkeiten zu veranschaulichen. Dabei werden Diagramme zum Darstellen von Anwendungsfällen und fachlicher Workflows verwendet. UML-Diagramme finden in der objektorientierten Modellierung Anwendung, wie auch Entity Relationship Diagramme im Datenbankentwurf. Auch in der Theoretischen Informatik werden Zeichnungen von Syntaxdiagrammen, Endlichen Automaten, Ableitungsbäumen, Rekursion und Divide & Conquer zur Veranschaulichung der Abläufe und Zusammenhänge angefertigt. Um den reibungslosen Ablauf in Firmen oder Produktionen sicherzustellen, geben Organisationspläne und Produktionspläne Abhängigkeiten wieder.

Des Weiteren werden Zeichnungen zur Herausstellung von Zusammenhängen auch in nichtfachlichen Bereichen wie in sozialen Netzwerken verwendet. Weitere Beispiele für die Notwendigkeit von Zeichnungen und Diagrammen sind die übersichtliche Darstellung firmeninterner Hierarchien sowie die Repräsentation von Webseiten und deren Verlinkungen. Unter der Annahme, dass themenähnliche Webseiten tendenziell stärker untereinander verlinkt sind, würde man von Hand zuerst nach Themen, wie z. B. Politik, Sport, Wissenschaft, Forschung u. v. m. sortieren, um diese dann in einen gemeinsamen Bereich in der Zeichnung zu platzieren. Hierbei stellt sich die Frage, wie solche Informationen ausgewertet und anschließend lesbar präsentiert werden können.

Traditionell wurden und werden Zeichnungen kleiner Datenmengen von Hand erstellt. Daten werden dabei als Punkte, Kreise, Rechtecke oder als andere geometrische Formen dargestellt. Die Abhängigkeiten oder die Verbindungen zwischen Paaren von Datensätzen werden als Linien eingezeichnet. Der Ersteller kann dabei die Struktur der vorgegebenen Daten in der Zeichnung wiedergeben, um die Verständlichkeit für die späteren Betrachter zu erhöhen. Das konkrete Vorgehen ist keinen strengen Regeln unterworfen, so dass es den subjektiven Empfindungen des Erstellers obliegt, was er für „schön“ und „verständlich“ hält. Anhand dieser subjektiven Kriterien kann er selbst entscheiden, wie er Objekte und deren Beziehungen in der Zeichnung zueinander platziert und wie die Verbindungen zwischen Objekten verlaufen.

Ein einfaches Beispiel der verschiedenen subjektiven Empfindungen ist in Abbildung 1.1 gezeigt, wo jeweils derselbe Graph mit vier Knoten und Verbindungen zwischen allen Knotenpaaren auf unterschiedliche Art und Weise dargestellt ist. In Abbildung 1.1(a) ist eine Zeichnung des Graphen mit geradlinigen Verbindungslinien gezeigt. Aufgrund der Kreuzung zweier Verbindungen könnte man anstreben, diese Kreuzung zu eliminieren ohne die Knotenpositionen zu verändern, was zu Abbildung 1.1(b) führt. Es ist jedoch auch möglich durch Verschieben der Knoten eine Zeichnung zu erstellen, deren Verbindungen geradlinig gezeichnet werden und zudem keine Kreuzungen enthält, siehe Abbildung 1.1(c). Des Weiteren haben die Zeichnungen in Abbildung 1.1 verschiedene Symmetriegehalte. Die geradlinige Zeichnung mit einer Kantenkreuzung in Abbildung 1.1(a) ist symmetrischer gezeichnet als die Zeichnung ohne Kantenkreuzung in Abbildung 1.1(b). Welche dieser drei Zeichnungen die „schönste“ ist, muss individuell entschieden werden. Abhängig davon, was mit der Zeichnung dargestellt werden soll, kann man eine dieser drei Zeichnungen als die „schönste“ bewerten. In diesem Fall kann für und gegen jede dieser drei Zeichnungen argumentiert werden, so dass sich objektiv keine schönste Zeichnung bestimmen lässt. Dazu müsste man Kriterien festlegen, die eine Zeichnung erfüllen muss oder soll.

Möchte man beispielsweise den Informationsfluss von Daten darstellen, so soll dieser Fluss aus der Zeichnung ablesbar sein. Werden Daten oder Informationen nacheinander über verschiedene Stationen gesendet, so ist es erstrebenswert, dass der zeitliche Ablauf in der Zeichnung schnell ersichtlich ist. Die Richtung könnte entsprechend der üblichen Leserichtung von links nach rechts festgelegt werden.



**Abbildung 1.1.** Drei mögliche Zeichnungen Graphen  $G$

Auch der Produktionsplan von Automobilen könnte von links nach rechts repräsentiert werden, wobei gerichtete Verbindungen zwischen Produktionsschritten eingefügt werden, wenn ein Schritt zwingend vor einem anderen ausgeführt werden muss. Ein Beispiel dafür wäre, dass ein Rad erst hergestellt werden muss, bevor es am Automobil festgeschraubt werden kann oder, dass eine Front- oder Heckscheibe erst konstruiert werden muss, bevor man sie in die Karosserie einsetzen kann. Ein anderes Beispiel für gerichtete Zeichnungen ist das Darstellen firmeninterner Hierarchien, wo festgelegt ist, welche Personen anderen Personen Weisungen oder Arbeiten auftragen dürfen. Aus der vorgegebenen Rangordnung würde man eine Person von hohem Rang in einer Zeichnung entsprechend weiter oben platzieren, und umgekehrt.

Das Erstellen von Zeichnungen großer Datenmengen von Hand ist problematisch, da dies sehr zeitaufwändig und mühsam ist. Nicht nur die Anzahl der resultierenden Objekte in Zeichnungen erschweren die Erstellung, sondern auch die Frage, wo die Objekte platziert werden. Mit Stift und Papier ist es sehr knifflig den Objekten endgültige Positionen zuzuweisen. Aus diesen Gründen hat sich das Forschungsgebiet des automatisierten Zeichnens von Daten und deren Zusammenhänge, das sogenannte *Graph Drawing*, entwickelt. Die Anfänge des Graphenzeichnens wurden mitunter durch die Frage von Knuth „How shall we draw a tree?“ begründet [Knu68]. Bereits in einfachen Beispielen zeigt sich, wie schwierig es ist, zu bestimmen, welche der Zeichnungen die übersichtlichste oder lesbarste ist, siehe Abbildung 1.1. Will man Kreuzungen in Zeichnungen vermeiden? Sollen die Verbindungen geradlinig gezeichnet werden oder werden Knicke in den Verbindungen erlaubt? Will man beide Eigenschaften in einer Zeichnung vereinen? Sollen die Verbindungen bestimmte Richtungen besitzen? Sollen die Verbindungen nur horizontal und vertikal verlaufen? Ist erforderlich, bestimmte Teile in der Zeichnung nicht im selben Bereich zu platzieren? Diese Fragen sind nur ein kleiner Auszug aus den möglichen Fragestellungen, die sich beim Zeichnen von Graphen ergeben.

Um automatisiert Zeichnungen erstellen zu lassen, müssen zuerst sinnvolle Kriterien definiert werden, wodurch sich die Qualität verschiedener Darstellungen messen lässt. Grundlegende Definitionen, Darstellungsformen und deren Qualität werden im

folgenden Kapitel 2 festgelegt und für verschiedene Klassen von Graphen bewertet und diskutiert. In Kapitel 3 werden Baumzeichnungen eingeführt sowie verschiedene Problemstellungen aufgegriffen und diskutiert. Des Weiteren werden neue Ansätze präsentiert und deren Komplexität bestimmt. Danach werden in Kapitel 4 Verfahren zum Zeichnen von Bäumen auf Gittern beschrieben. In Kapitel 5 wird eine Zusammenfassung und ein Ausblick für mögliche zukünftige Arbeiten gegeben.

# 2

## Einführung

---

In diesem Abschnitt werden grundlegende Definitionen und Ansätze des Zeichnens von Graphen vorgestellt. Nach der Definition eines Graphen und einer Zeichnung eines Graphen werden Qualitätsmerkmale und Ansätze zum Zeichnen verschiedener Graphklassen angeführt. Qualitätsmerkmale sollen dabei sicherstellen, dass Zeichnungen automatisch und objektiv bewertet werden können, ohne dabei eine subjektive Meinung einfließen zu lassen.

**Definition 2.1 (Graph).** *Ein Graph  $G = (V, E)$  besteht aus einer endlichen Menge von Knoten  $V$  und einer endlichen Menge von ungerichteten Kanten  $E \subseteq \mathcal{P}_2(V)$  (gerichteten Kanten  $E \subseteq V \times V$ ). Eine ungerichtete Kante zwischen zwei Knoten  $u$  und  $v$  mit  $u, v \in V$  wird definiert durch das Paar  $\{u, v\}$ . Das entsprechende geordnete Paar  $(u, v)$  definiert eine gerichtete Kante [DBETT99, KW01].*

Sind alle Kanten von  $G$  gerichtet (ungerichtet) so ist  $G$  *gerichtet* (*ungerichtet*) [DBETT99, KW01]. Zwei Knoten  $u, v \in V$  eines Graphen  $G = (V, E)$  werden als *adjazent* bezeichnet, wenn eine Kante  $e = \{u, v\} \in E$  existiert. Die Knoten  $u$  und  $v$  sind *Nachbarn* und  $e$  ist *inzident* zu  $u$  und  $v$  und umgekehrt. Ein *Pfad* (*Weg*) in einem Graphen  $G = (V, E)$  ist eine Folge  $(v_1, v_2, \dots, v_h)$  von verschiedenen Knoten von  $G$  mit  $(v_i, v_{i+1}) \in E$  ( $\{v_i, v_{i+1}\} \in E$ ) und  $i \in \{1, \dots, h-1\}$ . Die Länge  $l(p)$  des Pfades oder Weges  $p$  ist die Anzahl der Kanten von  $p$ .

Ein Graph  $G$  abstrahiert dabei von den Daten auf die Knoten und deren zweistelligen Beziehungen auf die Kanten. Die beiden Mengen  $V$  und  $E$  ermöglichen einfaches Speichern in Adjazenzlisten oder Adjazenzmatrizen [AHU74, DBETT99,

**KW01**]. Diese Arten des Speicherns haben den Nachteil, dass strukturelle Informationen nicht einfach abzulesen sind. Es wird das Beispiel aus Abbildung 1.1 aufgegriffen. Die Darstellung des Graphen  $G$  in Mengenschreibweise ist:

$$G = (V, E) = (\{v_1, v_2, v_3, v_4\}, \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}, \{v_1, v_3\}, \{v_2, v_4\}\})$$

In dieser mengentheoretischen Darstellung ist nicht leicht abzulesen, dass alle Knotenpaare durch eine Kante verbunden sind. In einer Adjazenzmatrix für  $G$  ergibt sich die folgende Matrix:

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Dabei bedeutet eine 1 in der  $i$ -ten Zeile und der  $j$ -ten Spalte, dass es in  $G$  eine Kante zwischen Knoten  $v_i$  und  $v_j$  gibt, andernfalls gibt es für eine 0 keine Kante zwischen  $v_i$  und  $v_j$ . Daran lässt sich leichter erkennen, dass der Graph *vollständig* ist, da für jedes Knotenpaar eine Kante existiert. In der Adjazenzlistendarstellung würde  $G$  folgendermaßen dargestellt:

$$v_1 : \{v_2, v_3, v_4\} \tag{2.1}$$

$$v_2 : \{v_1, v_3, v_4\} \tag{2.2}$$

$$v_3 : \{v_1, v_2, v_4\} \tag{2.3}$$

$$v_4 : \{v_1, v_2, v_3\} \tag{2.4}$$

Für jeden Knoten  $v_i \in V$  wird die Menge seiner benachbarten Knoten gespeichert. Beispielsweise hat  $v_1$  die Nachbarn  $v_2, v_3$  und  $v_4$ . Anhand dieses einfachen Beispiels kann man sich schnell vergegenwärtigen, wie die Struktur des Graphen aussieht. Jedoch ist es bei größeren Graphen mit mehr Knoten und mehr Kanten sinnvoll die zeichnerische Repräsentation eines Graphen zu wählen, aus der man in der Regel schneller einen Einblick in die Struktur der zu repräsentierenden Daten bekommt.

## 2.1 Zeichnung eines Graphen

Damit man möglichst auf einen Blick die Grundstruktur eines Graphen  $G = (V, E)$  erkennen kann, bietet sich eine graphische Darstellung an. So wird es dem Betrachter bei geeigneter Repräsentation erleichtert, strukturelle Informationen effizient abzulesen.

**Definition 2.2 (Einbettung, Zeichnung).** Eine Einbettung  $\gamma(G)$  eines Graphen  $G = (V, E)$  weist jedem Knoten  $v \in V$  eine Reihenfolge  $\pi : E(v) \rightarrow \mathbb{N}$  seiner adjazenten Knoten um  $v$  zu. Eine Zeichnung eines Graphen  $G = (V, E)$  ist gegeben durch eine Funktion

$$\Gamma : \begin{cases} V \mapsto \mathbb{R}^2, \\ E \mapsto \{J : [0, 1] \mapsto \mathbb{R}^2 \mid J \text{ ist eine Jordankurve}\} \end{cases}$$

mit der Eigenschaft  $J = \Gamma(u, v)$  und

- $J$  ist stetig und injektiv (Definition der Jordankurve)
- $J(0) = \Gamma(u)$  und  $J(1) = \Gamma(v)$ .

Die Fläche einer Zeichnung  $\Gamma(G)$  eines Graphen  $G$  ist definiert als das kleinste umschließende Rechteck  $A = H \cdot W$  mit Höhe  $H$  und Breite  $W$ . Wie bereits erwähnt, soll ein Graph in einer Zeichnung so dargestellt werden, dass die Kriterien der Verständlichkeit, Übersichtlichkeit, Lesbarkeit und auch „Schönheit“ im Optimalfall gleichzeitig beachtet werden. Diese sehr subjektiven Kriterien können jedoch nicht für das automatische und objektive Auswerten von Zeichnungen verwendet werden. Jeder Mensch als Individuum würde diese Kriterien, wenn auch nur geringfügig, anders definieren. Und jedes Verfahren braucht klare Definitionen nach denen es entscheiden kann, welche Knotenpositionen vergeben werden. Deshalb ist es notwendig, objektive Kriterien zu finden und festzulegen. Zwei solche Kriterien wären beispielsweise, dass Knoten einen gewissen Mindestabstand in der Zeichnung einhalten müssen und dass sich Kanten zwischen zwei Knoten nicht schneiden dürfen. Wird ein Mindestabstand zwischen Knoten nicht gefordert, können Knoten so nahe nebeneinander platziert werden, so dass die Differenzierung erheblich erschwert wird. Ähnlich ist es mit Kantenkreuzungen. Schneiden sich viele Kanten, so ist es nicht leicht zu erkennen, welchen Knoten adjazent sind. Hierbei ist zu bemerken, dass nicht jeder Graph ohne Überschneidungen von Kanten dargestellt werden kann [Kur30], so dass sich die Forderung nach möglichst wenigen Kantenkreuzungen aufdrängt.

Die folgenden Beispiele sollen belegen, dass es keine allgemeingültigen Kriterien gibt, mit denen man die Übersichtlichkeit von Zeichnungen messen kann. Sei der Graph  $G_{prod} = (V_{prod}, E_{prod})$  ein Graph mit  $V_{prod}$  einer Menge von Produktionsschritten (z. B. Konstruktion in Automobilbranche) und den Kanten  $E_{prod}$ , die festlegen ob ein Produktionsschritt nach einem anderen ausgeführt werden muss. Ein weiterer Graph  $G_{hier} = (V_{hier}, E_{hier})$  soll eine firmeninterne Hierarchie darstellen. Dabei repräsentiert die Menge der Knoten  $V_{hier}$  die in der Firma beschäftigten Personen und die gerichteten Kanten  $E_{hier}$  legen fest, welche Personen die Vorgesetzten von anderen Personen sind. Hierbei wird angenommen, dass jede Person höchstens einen Vorgesetzten hat, aber ein Vorgesetzter mehreren Personen weisungsbefugt ist. Stellt man sich die Aufgabe  $G_{prod}$  und  $G_{hier}$  zu zeichnen, so müssen sinnvolle Kriterien gefunden werden, um eine übersichtliche Zeichnung zu erstellen.

Beide Graphen sind gerichtet, was aber nicht bedeutet, dass sie beide nach den gleichen Kriterien gezeichnet werden sollen. Der Produktionsablauf würde wohl entlang einer Zeitleiste (beispielsweise von links nach rechts) gezeichnet werden, um den Fortschritt der Konstruktion ablesen zu können. Die Hierarchie in einer Firma hingegen könnte durch das Zeichnen von oben nach unten dargestellt werden. Der Chef der Firma steht ganz oben und unterhalb werden die Personen aufgelistet, denen der Chef direkt weisungsbefugt ist, usw. Für beide Graphen werden vom Menschen intuitiv verschiedene Ansätze verwendet.

Diese verschiedenen Ansätze sind notwendig, um die Eigenschaften der Graphen sinnvoll und übersichtlich darzustellen. Sind diese Informationen bekannt, so kann man sich aus verschiedenen Darstellungsformen für eine entscheiden, um ein möglichst gutes Ergebnis zu erhalten. Dieses Ergebnis soll die Struktur lesbar und übersichtlich wiedergeben. Solche Bewertungskriterien für Zeichnungen von Graphen verschiedener Klassen werden im nächsten Abschnitt 2.2 eingeführt und diskutiert.

## 2.2 Qualität von Zeichnungen

Damit man Zeichnungen von Graphen in ihrer Qualität bewerten kann, werden im Folgenden Eigenschaften aufgelistet, die für eine gesamte Zeichnung gelten müssen und/oder gelten können. Dabei kommt es maßgeblich darauf an, dass die Kriterien objektiv und automatisch bewertet werden können. Zuletzt werden außerdem Kriterien angegeben, die lediglich für einen Teil der Zeichnung festgelegt werden können und somit nicht für die komplette Zeichnung erfüllt werden müssen.

### 2.2.1 Zeichenkonventionen

Zunächst werden harte Vorgaben definiert, die in der gesamten Zeichnung eines Graphen eingehalten werden müssen.

**Definition 2.3 (Zeichenkonvention).** *Eine Zeichenkonvention ist eine global für die Zeichnung geltende Eigenschaft, die eine Zeichnung erfüllen muss. Nur wenn alle geforderten Zeichenkonventionen für eine Zeichnung erfüllt sind, ist eine Zeichnung zulässig [DBETT99].*

Für verschiedene Typen von Graphen ergeben sich unterschiedliche Zeichenkonventionen. Beispielsweise wird oft die Ganzzahligkeit der Knotenpositionen in einer Zeichnung gefordert [Shi76, Woo81, SR82]. Es ist aber auch üblich Knoten als geometrische Formen, wie Rechtecke, Kreise o. ä. darzustellen, um die Beschriftung innerhalb dieser Formen zu platzieren. Dann könnte eine Zeichenkonvention lauten, dass sich die geometrischen Formen verschiedener Knoten nicht überlagern dürfen. Die gezeichneten Kanten können dabei ebenfalls verschiedene Eigenschaften besitzen. Eine Zeichnung ist *polyline*, wenn eine Kante als Polylinie gezeichnet ist. Ist jede Kante in einer Zeichnung geradlinig, so wird die Zeichnung *geradlinig* genannt, siehe Abbildung 1.1(a) oder 1.1(c). Verlaufen in einer Zeichnung alle Kanten

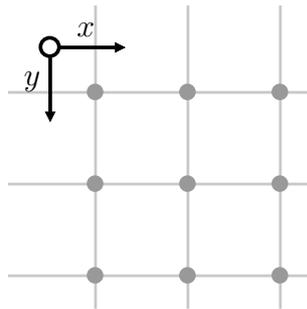


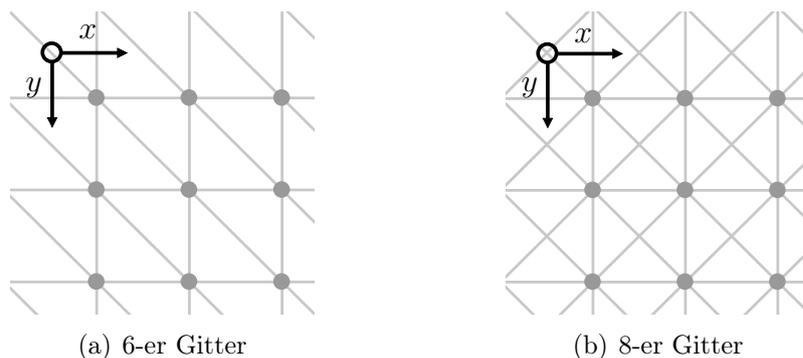
Abbildung 2.1. 4-er Gitter

(streng) monoton in eine bestimmte Richtung, so wird sie eine (*strenge*) *Aufwärtszeichnung* genannt. Diese Zeichenarten sind hauptsächlich in azyklischen Graphen vorteilhaft, um Abläufe oder auch hierarchische Strukturen [STT81, Tre96] darzustellen. Eine weitere Zeichenkonvention ist die Forderung nach *Planarität*, d. h. keine Überlagerung oder Kreuzung von Kanten und/oder Knoten. Eine Zeichnung ist *planar*, wenn keine Überlagerungen zwischen Knoten und/oder Kanten existieren. Ein Graph ist planar, wenn eine planare Zeichnung existiert. Mit Hilfe von Planaritätstests kann in Linearzeit bestimmt werden, ob ein Graph planar gezeichnet werden kann [LEC67, HT74, CNAO85]. Ist ein Graph planar, so kann er mit dem FPP-Algorithmus planar gezeichnet werden [DFPP90]. Die mit diesem Verfahren gezeichneten Graphen besitzen keine Kantenkreuzungen und verwenden ganzzahlige Koordinaten. Jedoch sind die Winkel zwischen inzidenten Kanten gegebenenfalls sehr klein, so dass diese nicht mehr leicht differenziert werden können. Die *Winkelauflösung* in einer Zeichnung eines Graphen ist der kleinste Winkel zwischen inzidenten Kanten. In dieser Arbeit werden Zeichnungen mit einer festen Winkelauflösung betrachtet. Deren Knoten und Kanten werden in ein vorliegendes Gitter eingebettet, so dass die Winkelauflösung bereits vorher durch die Richtungen der Kanten des unterliegenden Gitters bestimmt ist.

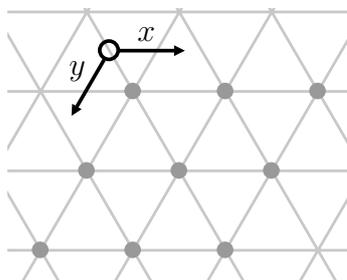
**Definition 2.4 (Gitter, Gittergraph).** Sei  $G = (V, E)$  ein endlicher Graph und  $\Gamma : V \mapsto \{1, \dots, n\} \times \{1, \dots, m\}$  für  $n, m \in \mathbb{N}$  eine Bijektion. Für  $i < n$  und  $j \leq m$  existiere für jeden Knoten  $v_{i,j}$  eine Kante zwischen  $v_{i,j}$  und  $v_{i+1,j}$  und für  $i \leq n$  und  $j < m$  eine Kante zwischen  $v_{i,j}$  und  $v_{i,j+1}$ . Dann wird  $G$  als Gitter oder Gittergraph bezeichnet. Die Kanten eines Gitters werden Gitterkanten genannt.

Ein Gittergraph  $G = (V, E)$  wird im Folgenden auch als *orthogonales Gitter* oder *4-er Gitter* bezeichnet. In einem 4-er Gitter hat jeder Knoten exakt vier adjazente Knoten, siehe Abbildung 2.1.

Erweitert man das 4-er Gitter aus Definition 2.4, indem man für jeden Knoten  $v_{i,j}$  eine Kante zum Knoten  $v_{i+1,j+1}$  einfügt, so erhält man das *hexagonale Gitter* bzw. *6-er Gitter* mit sechs adjazenten Knoten für jeden Knoten  $u \in V$ . Das *oktagonale Gitter* bzw. *8-er Gitter* ergibt sich, wenn man zum hexagonalen Gitter für



**Abbildung 2.2.** Erweiterungen des üblichen orthogonalen Gitters



**Abbildung 2.3.** Symmetrisches 6-er Gitter

jeden Knoten  $v_{i,j}$  eine Kante zum Knoten  $v_{i+1,j-1}$  einfügt. Das 6-er Gitter ist in [Abbildung 2.2\(a\)](#) und das 8-er Gitter in [Abbildung 2.2\(b\)](#) gezeigt.

In der Zeichnung des 6-er Gitters in [Abbildung 2.2\(a\)](#) kann man erkennen, dass die drei verschiedenen Gitterachsen Winkel von  $\pi/2$  und  $\pi/4$  Grad einschließen und somit nicht gleich groß sind. Um den Symmetriegehalt der Zeichnung zu erhöhen, können die Achsen mittels einer Lineartransformation rotiert werden, so dass sich das *symmetrische 6-er Gitter* ergibt, siehe [Abbildung 2.3](#). Dabei wird die diagonale Achse um  $\pi/12$  und die y-Achse um  $\pi/6$  im Uhrzeigersinn gedreht. Das hexagonale Gitter kann somit entweder auf die letztere Art und Weise symmetrisch gezeichnet werden.

Die symmetrische Variante liefert aus ästhetischer Sicht schönere Zeichnungen, da die Winkel einheitlich sind und die Winkelauflösung  $\pi/3$  statt  $\pi/4$  ist. Diese Lineartransformation ermöglicht es, dass Berechnungen entweder auf den Koordinaten des 6-er Gitters oder des symmetrischen 6-er Gitter ausgeführt werden können, da nach der Berechnung der einen oder anderen Zeichenart die Koordinaten in Linearzeit transformiert werden können.

Im Folgenden werden das 4-er, 6-er und 8-er Gitter zusammengefasst als *k-Gitter* mit  $k \in \{4, 6, 8\}$  bezeichnet.

**Definition 2.5 (Gittereinbettung).** Eine Gittereinbettung  $\Gamma_k(G)$  eines Graphen  $G = (V, E)$  in ein  $k$ -Gitter  $G_k = (V_k, E_k)$  ist eine injektive Abbildung  $\gamma$  von  $V$  auf  $V_k$  sowie eine injektive Abbildung von  $E$  auf die Menge der Pfade von Gitterkanten aus  $E_k$ , wobei jede Kante  $e = (u, v) \in E$  auf einen Pfad beginnend bei  $\gamma(u)$  und endend bei  $\gamma(v)$  abgebildet wird.

Ist ein Graph in ein Gitter eingebettet, so wird der Abstand zwischen zwei Knoten in dieser Arbeit nicht auf die übliche Weise mit dem euklidischen Abstand bestimmt, sondern mit der Maximumsmetrik. Ein Nachteil der euklidischen Abstandsberechnung ist nämlich, dass dadurch reelle Zahlen für Berechnungen benötigt würden.

**Definition 2.6 (Distanz, Abstand).** Die Distanz oder der Abstand  $d(u, v)$  zwischen zwei Knoten  $u$  und  $v$  auf den ganzzahligen Koordinaten  $(x_u, y_u)$  und  $(x_v, y_v)$  wird durch die Maximumsmetrik bestimmt:

$$d(u, v) = d((x_u, y_u), (x_v, y_v)) = \max(|x_u - x_v|, |y_u - y_v|).$$

Betrachte eine Gittereinbettung  $\Gamma_k(G)$  eines Graphen  $G = (V, E)$  in ein  $k$ -Gitter, d. h. Knoten und Kanten sind in das  $k$ -Gitter eingebettet. Haben alle Kantensegmente einer Kante, die auf einen Pfad in  $\Gamma_k(G)$  eingebettet sind, die gleiche Richtung, so ist die Kante *geradlinig* in das Gitter eingebettet, siehe Abbildung 2.4(a). Beispiele für nicht-geradlinige Gittereinbettungen sind in Abbildung 2.4(b) zu sehen. Werden nur die Knoten auf das Gitter eingebettet und die Kanten bzw. Kantensegmente nicht auf die Kanten des Gittergraphen eingebettet, so nennt man die entsprechenden *Kanten nicht in das Gitter eingebettet*, siehe Abbildung 2.4(c) und Abbildung 2.4(d). In dieser Arbeit werden die Knicke sowie die Knoten ausschließlich auf ganzzahligen Koordinaten platziert.

Bei den oben genannten Kriterien handelt es sich um Zeichenkonventionen, die von einer Zeichnung gefordert werden können. Diese Konventionen sind harte Kriterien, die eine gültige Zeichnung erfüllen muss. Werden diese Kriterien nicht erfüllt, so ist eine Zeichnung nach Definition 2.3 ungültig.

### 2.2.2 Ästhetikkriterien

Die harten Kriterien im vorigen Abschnitt stellen Musskriterien für Zeichnungen dar. In diesem Teil werden Kriterien angeführt, die in einer Zeichnung meist nicht exakt eingehalten werden können. Sie können nur bis zu einem gewissen Grad erreicht werden und können deshalb mit Wunschkriterien verglichen werden. Ein Beispiel hierzu ist das Zeichnen eines Graphen mit so wenig Kantenkreuzungen wie möglich.

Es ist sinnvoll „weiche“ Kriterien zusätzlich zu den Zeichenkonventionen zu verwenden, um die Qualität von Zeichnungen zu messen und diese gegebenenfalls zu verbessern.

**Definition 2.7 (Ästhetikkriterium).** Ein Ästhetikkriterium ist eine Eigenschaft

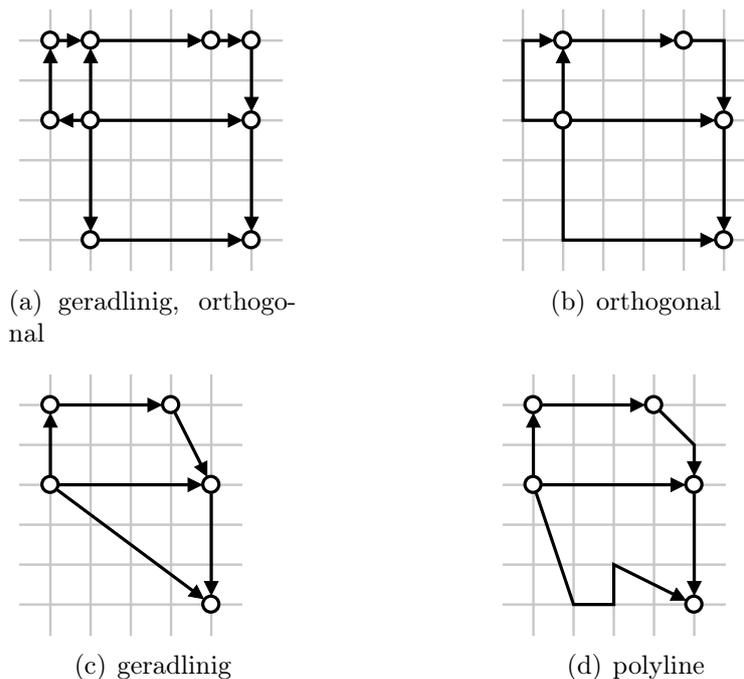


Abbildung 2.4. Gitterzeichnungen

einer Zeichnung, die, soweit wie möglich, erreicht werden soll, um die Lesbarkeit und die Übersichtlichkeit zu erhöhen [DBETT99].

Hierbei ist zu beachten, dass sinnvolle Ästhetikkriterien für eine Zeichnung eines Graphen größtenteils von der Struktur eines Graphen abhängen. Des Weiteren gibt es keine klaren Regeln, die vorgeben, wie verschiedene Ästhetikkriterien zueinander gewichtet werden sollen [KW01, KMS94]. Aber es gibt viele Ansätze, welche die Lesbarkeit erhöhen. Ein wichtiges Ästhetikkriterium ist beispielsweise die Minimierung der Anzahl der Kantenkreuzungen. Wenn es in einer Zeichnung zu viele Kantenkreuzungen gibt, dann kann das menschliche Auge nicht mehr einfach nachvollziehen, welche Knoten durch welche Kanten miteinander verbunden sind. In Zeichnungen von nicht-planaren Graphen, die beispielsweise in Schaltkreisen vorkommen, ist es wichtig die Anzahl der Kreuzungen zu minimieren. In jeder Schicht können Leitungen nur kreuzungsfrei gezeichnet werden, so dass die Anzahl der Schichten von der Anzahl der Kreuzungen abhängig ist. Ob ein Graph ohne Kreuzungen gezeichnet werden kann, ist in  $\mathcal{O}(n)$  Zeit bestimmbar [LEC67, HT74, CNAO85]. Dem steht aber das Problem der Minimierung der Anzahl von Kreuzungen, das im Allgemeinen  $\mathcal{NP}$ -hart ist, gegenüber [GJ90].

Aufgrund der in der Praxis verwendeten Ausgabemedien, wie z. B. Bildschirme oder Druckerpapier, wird die Fläche in der Regel mit dem kleinsten umschließenden Rechteck gemessen. Damit auf Ausgabemedien sinnvolle Zeichnungen generiert werden können, ist es notwendig, die Zeichnungen auf kleiner Fläche zu erstellen.

Dabei muss auch das Seitenverhältnis beachtet werden. Eine Zeichnung mit einem sehr großen Seitenverhältnis kann nicht einfach auf einem Bildschirm dargestellt werden, ohne dabei ein Scrollen zu erzwingen. Hierbei ist ebenfalls zu beachten, dass es nicht sinnvoll ist, die Zeichnung beliebig zu skalieren, um einen guten Flächenwert zu erhalten. Haben Knoten in der Darstellung des Ausgabemediums einen sehr kleinen Abstand, kann man mit bloßem Auge nicht mehr unterscheiden, ob es ein oder zwei Knoten sind. Weitere Kriterien zur Verbesserung der Lesbarkeit werden im Folgenden aufgeführt:

- Summe aller Kantenlängen: Dieses Kriterium summiert alle Längen der Kanten auf. Vergleicht man zwei Zeichnungen, bei denen dieser Wert in einer viel größer ist, als in der anderen, dann kann man annehmen, dass die Qualität der Zeichnung mit dem höheren Wert schlechter ist. Je länger eine Kante in einer Zeichnung ist, desto schwerer ist es für das menschliche Auge, die Zusammengehörigkeit adjazenter Knoten schnell zu erkennen. Ein weiterer Vorteil dieses Kriteriums in Bezug auf Leitungen in Computerchips, sind die geringeren Kosten, die ein Chip-produzierendes Unternehmen minimieren möchte.
- Maximale Kantenlänge: Des Weiteren ist es oft erstrebenswert, die maximale Kantenlänge zu minimieren. Betrachtet man die Durchlaufzeit von Informationen durch Leitungen, so ist die Laufzeit durch die längste Leitung begrenzt.
- Uniforme Kantenlänge: Dieses Kriterium fordert eine kleine Varianz über alle Kantenlängen. Eine Zeichnung mit sehr unterschiedlichen Kantenlängen erscheint unregelmäßig und weniger attraktiv, wie eine Zeichnung mit kleiner Varianz über die Kantenlängen.
- Minimieren der Anzahl von Knicken: Das Verfolgen einer geradlinigen Kante ist mit dem Auge relativ einfach möglich. Haben Kanten jedoch viele Knicke, so erschwert es dem Betrachter die entsprechenden inzidenten Knoten zu erkennen. Deshalb ist es oft notwendig die Gesamtzahl der Kantenknicke in Zeichnungen zu minimieren.
- Minimieren der maximalen Anzahl von Knicken pro Kante: Ebenso wie beim vorherigen Kriterium möchte man die maximale Anzahl der Kantenknicke pro Kante minimieren. Zwar ist es in vielen Fällen sinnvoll Kantenknicke zuzulassen. Jedoch möchte man für jede Kante eine Maximalzahl angeben, die nicht überschritten werden darf.
- Winkelauflösung: Die Winkelauflösung eines Graphen ist der Wert des kleinsten Winkels den zwei inzidente Kanten einschließen. Bei einem sehr kleinen Winkel kann man gegebenenfalls nicht mehr unterscheiden, ob es sich um eine oder um zwei Kanten handelt. Folglich soll die Winkelauflösung in Zeichnungen möglichst maximiert werden.

- Fläche: Die Fläche  $A$  einer Zeichnung soll minimiert werden. Dieses Kriterium ist vor allem im VLSI-Design (*very-large-scale integration*), das in der Chip-Industrie Anwendung findet, entscheidend. Das Fordern dieses Kriteriums hat zur Folge, dass auf kleiner Fläche die Knoten einer Zeichnung das Ausgabemedium gleichmäßiger und dichter ausfüllen.
- Seitenverhältnis: Das Seitenverhältnis der Fläche  $A = H \cdot W$  einer Zeichnung, die in Höhe  $H$  und Breite  $W$  angegeben wird, soll möglichst quadratisch sein. Grund dafür sind die Normen der üblichen Ausgabemedien, wie z. B. Computerbildschirm, Fernseher, Papier (DIN A4) oder auch Computerchips.
- Symmetrie: Wenn Graphen Symmetrien enthalten, d. h. ähnliche bis gleiche Strukturen, so sollen diese ebenfalls ähnlich oder gleich gezeichnet werden [LNS85]. Hier ist einerseits Achsen- oder Punktsymmetrie gemeint und andererseits die Rotationssymmetrie. Werden symmetrische Teile eines Graphen ähnlich gezeichnet, so können diese durch den Betrachter schneller wiedererkannt werden.
- Konvexe Flächen: Als ein weiteres Kriterium wurde gefordert, dass sich die aus einer planaren Zeichnung ergebenden durch die Kanten begrenzten Flächen konvex sein sollen [CON85].

Aus den angeführten Ästhetikkriterien lässt sich erkennen, dass man nicht alle Kriterien gleichzeitig in einer Zeichnung erfüllen kann. Manche Kriterien widersprechen sich in gewisser Weise. Dies wird in Abbildung 1.1 sehr deutlich. Dort besteht ein Konflikt zwischen einer guten Symmetrieeigenschaft in Abbildung 1.1(a) gegen Planarität in Abbildung 1.1(b). Aus diesem Grund muss man abwägen, welche Kriterien für eine Zeichnung wichtig bzw. unabdingbar sind. Solche Kriterien sollen dann die Qualität einer aus einem Verfahren resultierenden Zeichnung bewerten. Zeichenkonventionen und Ästhetikkriterien sind globale Forderungen, die eine Zeichnung gestellt werden.

### 2.2.3 Nebenbedingungen

Gibt es zu den Zeichenkonventionen und Ästhetikkriterien noch weitere Bedingungen, die Wissen über die Bedeutung einer Zeichnung voraussetzen, dann können weitere Kriterien definiert werden, die eine Zeichnung erfüllen muss. Diese semantischen Besonderheiten, die Wissen über die Bedeutung der Zeichnung beinhalten, können in *Nebenbedingungen* festgelegt werden [TDBB88, DBETT94].

Eine Nebenbedingung ist beispielsweise zu fordern, dass ein ausgezeichnete Knoten ins Zentrum einer Zeichnung platziert werden muss. Ein praxisrelevantes Beispiel hierzu ist die Repräsentation von zueinander in Beziehung stehender Personen in sozialen Netzwerken. Interessant ist, mit welchen Personen eine einzelne Person in Beziehung steht, so dass diese ins Zentrum der Zeichnung platziert wird. Als weitere Nebenbedingung könnte man fordern, dass eine Menge von Knoten in einem

gemeinsamen Bereich platziert werden soll. Analog können Eigenschaften für einzelne Kanten definiert werden, wie z. B. die Vorgabe einer bestimmten Richtung oder Länge.

## 2.3 Zeichenansätze verschiedener Graphklassen

Im Folgenden werden Zeichenverfahren für verschiedene Graphklassen mit bestimmten Eigenschaften diskutiert. Dabei werden zuerst Graphen angesprochen, über die keinerlei Informationen bekannt sind und Graphen, die beispielsweise einen Arbeitsablauf repräsentieren. Danach werden Graphen betrachtet, die ohne Kantenkreuzungen gezeichnet werden können.

### 2.3.1 Allgemeine Graphen

Ein *allgemeiner Graph* ist ein Graph, über den keine Strukturinformationen vorhanden sind, beispielsweise der Verlinkungsgraph im Web. Für allgemeine Graphen werden Verfahren, die an die Physik angelehnt sind, oder die Mehrdimensionale Skalierung verwendet [She62, Kru64, Ead84, KK89, FR91, WM95, DH96, KW01].

Bei den physikalischen *Spring Embedder* Verfahren werden Knoten als Metallringe und Kanten als Federn zwischen den Metallringen angesehen (siehe Abbildung 2.5). Für die Kanten in einer Zeichnung wird eine Ideallänge vorgegeben, die der Länge einer Feder im entspannten Zustand entspricht. Der Ausgangspunkt eines Spring Embedder Verfahrens ist eine zufällige Wahl der Knotenpositionen in einer Zeichnung oder die Vorgabe der Positionen durch den Benutzer. Aus dieser Startzeichnung ergeben sich die Abstände zwischen adjazenten und nicht adjazenten Knotenpaaren. Zwei adjazente Knoten mit einem Abstand, der größer ist als die festgelegte Ideallänge einer Kante, implizieren eine anziehende Kraft, welche physikalisch gesehen aus der Kraft einer gedehnten Feder resultiert. Im Gegensatz dazu stoßen sich zwei Knoten ab, die eine geringere Distanz als die Ideallänge besitzen.

Ein Spring Embedder Verfahren umfasst eine endliche Anzahl an Iterationen. In einer Iteration wird die Summe der Kräfte berechnet, die auf einen Knoten wirken. Dazu werden für jeden Knoten die Kräfte zu den restlichen Knoten berechnet und in einem Kraftvektor aufsummiert, um nachfolgend die Verschiebung des Knotens in dessen Richtung durchzuführen.

Es wird darauf abgezielt, dass Knoten gleichmäßig auf der zur Verfügung stehenden Zeichenfläche verteilt sind, und dass im Speziellen adjazente Knoten nicht zu weit voneinander entfernt platziert sind. Diese Forderungen haben unter anderem zur Folge, dass sich Kanten nicht über die gesamte Zeichenfläche erstrecken und so die Lesbarkeit beeinträchtigen.

Da mit den Kräften zwischen adjazenten Knoten noch nicht verhindert wird, dass zwei nicht adjazente Knoten eine sehr geringe Distanz besitzen und damit für das Auge sehr schwer unterscheidbar sind, wird eine weitere abstoßende Kraft

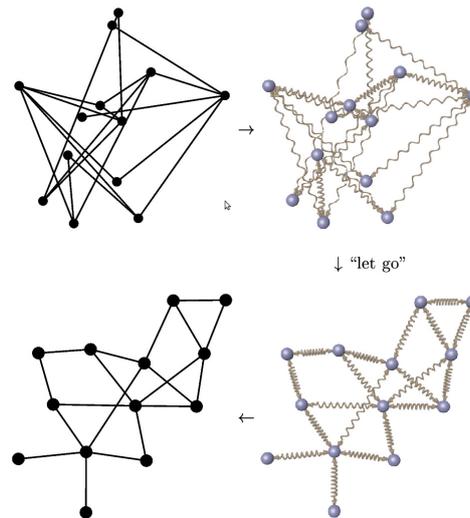


Abbildung 2.5. Spring Embedder (aus [KW01], Abb. 4.1)

zwischen allen Knotenpaaren eingeführt. Diese kann mit der Kraft zwischen zwei gleich geladenen Teilchen verglichen werden.

Die erste Idee eines Spring Embedder Verfahrens wurde 1984 von Eades präsentiert [Ead84]. Erweiterungen, wie insbesondere Federn zwischen allen Knotenpaaren, führten Kamada und Kawai [KK89] ein. Fruchterman und Reingold erweiterten die Idee von Eades, indem sie eine Temperatur simulieren, die von Iteration zu Iteration abfällt [FR91]. Dieser Ansatz gleicht dem Abkühlen eines flüssigen Stoffes, der in einen festen Zustand übergeht. Ebenso führten Fruchterman und Reingold eine Begrenzung der Zeichenfläche ein, so dass ein Auseinanderdriften des Graphen verhindert wird.

Ein Simulated Annealing Ansatz wurde von Davidson und Harel präsentiert, indem sie eine Funktion definierten, die abhängig von den individuell geforderten Ästhetikkriterien die Schönheit bzw. Lesbarkeit eines Graphens beschreibt [DH96]. Das Verfahren simuliert kein physikalisches System, sondern verschiebt die Knoten in den einzelnen Iterationen zufällig. Abhängig von einer von Iteration zu Iteration abfallenden Temperatur wird überprüft, ob sich die Lesbarkeit des Graphen anhand der festgelegten Funktion verbessert. Ist dies der Fall, so wird die Verschiebung durchgeführt. Hat die Verschiebung eine Verschlechterung zur Folge, so wird diese nur mit einer bestimmten Wahrscheinlichkeit durchgeführt.

### 2.3.2 Gerichtete (azyklische) Graphen

In allgemeinen Graphen stellen die Kanten Verbindungen ohne eine bestimmte Richtung dar. In Daten zu Produktionsabläufen oder auch Workflows ist jedoch meist eine Ablaufreihenfolge vorhanden, weshalb folgende Definitionen in der Graphen-

theorie Anwendung finden.

Ein gerichteter (ungerichteter) Graph enthält nur gerichtete (ungerichtete) Kanten. Eine *Schleife* ist eine Kante  $(u, v) \in E$  mit  $u = v \in V$  und eine *Mehrfachkante* ist eine Kante, die mehrfach in  $E$  vorkommt. Ein *einfacher Graph* ist ein Graph ohne Schleifen und ohne Mehrfachkanten. Im Folgenden wird angenommen, dass es sich bei allen Graphen um einfache Graphen handelt. Sei  $G = (V, E)$  und  $e = (u, v) \in E$  mit  $u, v \in V$ , dann ist  $e$  eine *ausgehende Kante* von  $u$  und zugleich eine *eingehende Kante* von  $v$ . In einem gerichteten Graphen hat jeder Knoten nur eingehende und ausgehende Kanten. Die Anzahl der ausgehenden Kanten  $\text{deg}^+(v)$  (eingehenden Kanten  $\text{deg}^-(v)$ ) eines Knotens  $v \in V$  bestimmt den *Ausgangsgrad* (*Eingangsgrad*) von  $v$ . Die Summe  $\delta(v) = \delta^+(v) + \delta^-(v)$  ist der *Grad* von  $v$ . Ein Pfad (Weg)  $(v_1, \dots, v_h)$  ist ein *Zyklus* (*Kreis*), wenn  $(v_h, v_1) \in E$  ( $\{v_h, v_1\} \in E$ ). Ein Graph ohne Zyklus (Kreis) ist ein *azyklischer* (*kreisfreier*) Graph. Ein Graph ist (*stark*) *zusammenhängend*, wenn zwischen allen Knotenpaaren mindestens einen Weg (Pfad) existiert. Ein *k-fach zusammenhängender* Graph ist ein Graph in dem zwischen allen Knotenpaaren  $k$  knotendiskunkte Wege zwischen allen Knotenpaaren existieren.

Sollen gerichtete Graphen gezeichnet werden, die einen Informationsfluss oder Workflow in der Zeichnung darstellen sollen, kann der physikalische Ansatz erweitert werden [SM95]. Zusätzlich zu den bereits bestehenden abstoßenden und anziehenden Kräften werden magnetische Kräfte simuliert, die auf Kanten wirken wie auf eine Kompassnadel im Magnetfeld. Hierbei können verschiedene Magnetfelder festgelegt werden, wie z. B. homogene oder radiale Magnetfelder. Das heute gängigste Algorithmenframework zum Zeichnen von gerichteten Graphen wurde von Sugiyama et al. [STT81] im Jahre 1981 erstmals vorgestellt. Bis heute wird das Framework für die Darstellung von gerichteten Graphen verwendet und laufend weiterentwickelt [JG72, BK01]. Das Framework beschreibt vier Phasen: Zykleneeliminierung, Ebeneneinteilung, Kreuzungsreduzierung und die abschließende Koordinatenzuweisung. Besitzen die Kanten eines Graphen eine Richtung, so ist es nicht ausgeschlossen, dass der Graph Zyklen enthält. Da es nicht möglich ist, alle Kanten eines Zyklus' in eine Richtung auszurichten, z. B. von links nach rechts, werden Zyklen zu Beginn aufgebrochen. Das bedeutet, die Kanten werden umgedreht oder vorübergehend gelöscht. Hierbei ist es nötig, möglichst wenig Kanten umzudrehen oder zu löschen, um die Struktur nicht unnötig stark zu verändern. Ist der Graph zyklensfrei, siehe Abbildung 2.6(a), so wird die zweite Phase ausgeführt. Es werden die Ebenen z. B. mit einer Heuristik von Coffman und Graham [JG72] festgelegt, auf denen die Knoten platziert werden, siehe Abbildung 2.6(b). In der dritten Phase werden dann die Reihenfolgen der Knoten auf den Ebenen bestimmt, um die Anzahl der Kreuzungen zu reduzieren. Die Reihenfolge der Knoten in Abbildung 2.6(b) in der untersten Ebene ist beispielsweise  $(L, M, O, N)$ . Nachdem jeder Knoten einer bestimmten Ebene zugeordnet ist und die Knoten auf jeder Ebene eine bestimmte Reihenfolge haben, werden abschließend die exakten  $x$ -Koordinaten der Knoten bestimmt [BK01].

Dieses Framework liefert Zeichnungen von gerichteten Graphen mit ganzzahligen Koordinaten. Die Anzahl der Kantenkreuzungen wurde in Phase 3 verringert und nach Phase 4 hat jede Kante höchstens zwei Knicke auf ganzzahligen Koordinaten.

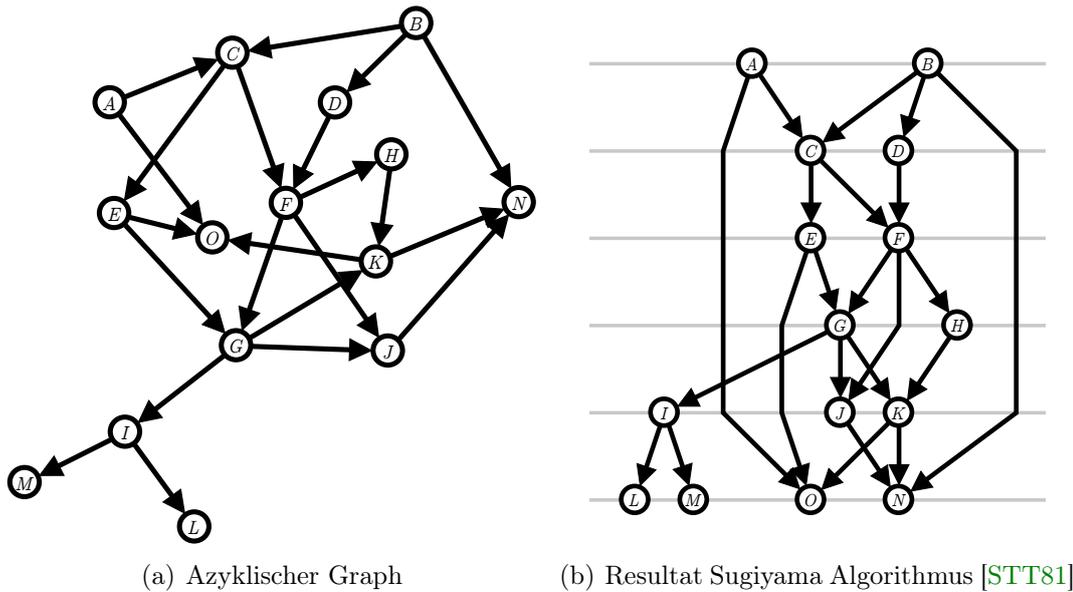


Abbildung 2.6. Azyklischer gerichteter Graph

Die Ergebnisse der einzelnen Phasen werden durch Heuristiken bewerkstelligt. In Phase 1, der Zyklenerliminierung, ist es das Ziel, möglichst wenig Kanten zu löschen oder umzudrehen. Das Problem, die minimale Anzahl zu finden, ist  $\mathcal{NP}$ -hart und auch unter dem Namen *Feedback-Arc-Set* bekannt [GJ90]. Ebenso ist es  $\mathcal{NP}$ -hart die Anzahl der Ebenen und gleichzeitig die Anzahl der Knoten auf den Ebenen zu minimieren [GJ90, UI175]. Die Minimierung der Anzahl von Kreuzungen ist ebenfalls  $\mathcal{NP}$ -hart [EK86].

### 2.3.3 Planare Graphen

Wie bereits im vorigen Abschnitt erläutert, soll, um die Lesbarkeit einer Zeichnung zu erhöhen, die Anzahl der Kantenkreuzungen minimiert werden. Ein Graph, der ohne Kantenkreuzungen gezeichnet werden kann, wird *planarer Graph* genannt. Es kann in Linearzeit getestet werden, ob ein Graph planar ist [LEC67, HT74, CNAO85]. Lempel, Even und Cederbaum präsentierten einen Ansatz, der ausgehend von einer *st*-Nummerierung, die eine Reihenfolge der Knoten festlegt, inkrementell Knoten hinzufügt [LEC67]. Aufgrund des inkrementellen Hinzufügens der Knoten, wird das Verfahren auch „vertex addition method“ genannt. Einen weiteren Planaritätstest stellten Hopcraft und Tarjan vor [HT74]. Dieser wird auch „path-addition method“ genannt, da er ausgehend von einem Zyklus, Schritt für Schritt Pfade hinzufügt. Beide Verfahren können in Linearzeit implementiert werden, wobei für das Verfahren von Lempel, Even und Cederbaum die Linearzeit von Booth und Lueker gezeigt wurde [BL76].

Für das Erstellen einer planaren Zeichnung eines planaren Graphens gibt es

verschiedene Algorithmen [Tut63, Woo81, DFPP90, Sch90]. Tutte zeigte, dass jeder 3-fach zusammenhängende planare Graph eine konvexe Zeichnung besitzt und präsentierte ein Verfahren, das eine planare geradlinige Zeichnung eines 3-fach zusammenhängenden Graphen erzeugt, deren Flächen (ohne die Außenfläche) konvex gezeichnet sind. *Konvex* bedeutet, dass jeder Winkel zwischen zwei inzidenten Kanten einer Fläche kleiner als  $\pi$  ist. Ein *maximal planarer Graph* ist ein planarer Graph, dessen Planarität durch das Hinzufügen einer beliebigen Kante zerstört wird. Der Algorithmus von de Fraysseix, Pach und Pollak berechnet ausgehend von einem maximal planaren Graphen zuerst eine kanonische Ordnung, um danach durch die inkrementelle Hinzunahme der Knoten, in der durch die kanonische Ordnung spezifizierten Reihenfolge, eine planare geradlinige Zeichnung auf ganzzahligen Koordinaten zu erzeugen [DFPP90]. Diese benötigt dabei die Fläche  $(2n - 4) \times (n - 2)$ . Chrobak und Payne konnten zeigen, dass das Verfahren in Linearzeit implementiert werden kann [CP95]. Schnyder präsentierte ebenfalls ein Linearzeitverfahren, das einen maximal planaren Graphen in der Fläche  $(n - 2) \times (n - 2)$  auf ganzzahlige Koordinaten geradlinig zeichnet. Wie auch das Verfahren von de Fraysseix, Pach und Pollak wird eine kanonische Ordnung berechnet, um damit die Koordinaten, unter Verwendung der baryzentrischen Repräsentation, zu bestimmen.

Ein Zeichenverfahren, das die Knoten und Kanten von Graphen mit einem maximalen Knotengrad vier in das orthogonale Gitter einbettet, wurde von Tamassia vorgestellt [Tam87]. Darin werden die Knoten und die Knicke der Kanten auf ganzzahlige Koordinaten platziert und die Kantensegmente verlaufen ausschließlich horizontal oder vertikal. Das Verfahren besteht aus drei Schritten. Zuerst wird die Topologie bestimmt, indem die Reihenfolge der Kanten um jeden Knoten ermittelt wird. Danach werden die Winkel zwischen inzidenten in der Einbettung aufeinanderfolgenden Kanten eines Knotens und die Winkel der Knicke berechnet, ohne dabei die Kantenlängen zu berücksichtigen, deren Berechnung im letzten Schritt vorgenommen wird.

Unter Verwendung der Resultate von de Fraysseix, Pach und Pollak [DFPP90] sowie Schnyder [Sch90] präsentierte Kant einen Linearzeitalgorithmus zum Zeichnen von 3-fach zusammenhängenden Graphen mit maximalem Knotengrad drei [Kan92]. Dieses erstellt Zeichnungen auf dem hexagonalen Gitter in der Fläche  $\frac{n}{2} \times \frac{n}{2}$ . Dabei hat höchstens eine Kante bis zu drei Knicke, wobei die restlichen Kanten geradlinig gezeichnet werden. Des Weiteren entwickelte er einen Algorithmus zum Zeichnen von 3-fach zusammenhängenden Graphen mit maximalem Knotengrad sechs. Dieser erstellt eine Zeichnung auf dem hexagonalen Gitter in der Fläche  $(6n - 12) \times (6n - 12)$ , so dass jede Kante höchstens vier Knicke besitzt.



# 3

## Zeichnen von Bäumen

---

Eine weitere Spezialisierung von allgemeinen, gerichteten und planaren Graphen sind Bäume. In diesem Kapitel werden Bäume, Eigenschaften von Bäumen sowie bekannte Verfahren und Darstellungsarten von Bäumen behandelt. Des Weiteren werden neue Ansätze für das Zeichnen von Bäumen auf Gittern präsentiert und deren Komplexität bezüglich der benötigten Fläche diskutiert.

**Definition 3.1 (Baum, geordnet, ungeordnet, Wurzelbaum).** *Ein Baum  $T = (V, E)$  oder auch freier Baum ist ein Graph, in dem es zwischen jedem Knotenpaar  $u, v \in V$  exakt einen Weg gibt. Ein Baum  $T$  ist geordnet, wenn die inzidenten Kanten eines Knotens  $v \in V$  einer Reihenfolge  $\pi(v)$  unterliegen. Dementsprechend ist die Reihenfolge der Kinder in einem ungeordneten Baum irrelevant. Gibt es in  $T$  einen Knoten  $r \in V$  der als Wurzel bezeichnet wird, so nennt man den Baum Wurzelbaum.*

Sei  $T_r = (V, E)$  ein Wurzelbaum mit Wurzel  $r \in V$ . Für jeden Knoten  $v \in V$  gibt es genau einen Weg  $w_v = (r = v_0, \dots, v_h = v)$  von der Wurzel  $r$  zum Knoten  $v$ . Hat jede Kante  $\{v_i, v_{i+1}\}$  jedes Weges  $w_v$  die Richtung  $(v_i, v_{i+1})$  mit  $i \in \{0, \dots, h(v) - 1\}$ , so wird der Wurzelbaum *gerichteter Wurzelbaum* genannt. Dabei ist die *Tiefe*  $t(v)$  von  $v$  mit  $v \in V$  die Länge des Pfades  $(r, \dots, v)$ .

In einem gerichteten Wurzelbaum  $T_r = (V, E)$  mit der Wurzel  $r$  hat jeder Knoten  $v \in V \setminus \{r\}$  genau eine eingehende Kante  $(u, v)$ , wobei  $u$  als der *Vater* von  $v$  bezeichnet wird. Die *Kinder*  $v_1, \dots, v_i$  von  $v$  mit  $i \in \mathbb{N}$  sind adjazente Knoten von  $v$ , die durch die gerichteten Kanten  $(v, v_1), \dots, (v, v_i)$  mit  $v$  verbunden sind. Der gerichtete Wurzelbaum  $T_r$  ist ein  $d$ -närer *Wurzelbaum*, wenn  $d = \max_{v \in V} \delta^+(v)$  gilt.

Demzufolge werden Bäume mit maximalen Ausgangsgrad  $d = 2$  *Binärbäume*,  $d = 3$  *Ternärbäume* oder auch  $d$ -näre Bäume genannt. In einem geordneten Binärbaum mit maximal zwei Kindern wird das erste Kind als *linkes Kind* und das Zweite als *rechtes Kind* bezeichnet. Ein Knoten  $u \in V$  ist *Vorfahre* von  $v \in V$ , wenn er auf dem Pfad von der Wurzel  $r$  zum Vater von  $v$  liegt. Dementsprechend ist  $v$  ein *Nachfahre* von  $u$ . Ein *Blatt* ist ein Knoten ohne Kinder und ein Knoten mit mindestens einem Kind ist ein *innerer Knoten*. Somit ist jeder Knoten entweder ein Blatt oder ein innerer Knoten. Die Knoten  $\{v \in V \mid t(v) = k\}$  sind Knoten des *Levels*  $k$ . Die *Höhe*  $h(T_r)$  ist definiert durch  $\max_{v \in V} \{t(v)\}$ . In einem *vollständigen  $d$ -nären gerichteten Wurzelbaum*  $T_r$  der Höhe  $h$  sind alle Level bis  $h$  voll besetzt, d.h. jeder Knoten vom ersten bis zum  $(h - 1)$ -ten Level hat genau  $d$  Kinder. Die Knoten des  $h$ -ten Levels, die Blätter, haben keine Kinder. Der *Unterbaum*  $T_v$  des Knotens  $v$  ist der Teilbaum der Knoten und Kanten mit  $v$  und all seinen Nachfahren mit den entsprechenden ausgehenden Kanten. Seien  $v_1, \dots, v_i$  die Kinder von  $v$ , dann werden die Knoten  $v_1, \dots, v_i$  *Geschwister* genannt.

Zum systematischen Ablaufen aller Knoten eines geordneten gerichteten Binärbaums  $T = (V, E)$  wird die *Inorder-Traversierung* definiert. Die Eingabe ist der ausgezeichnete Wurzelknoten  $r \in V$  und als Ausgabe wird jedem Knoten  $v \in V$  eine Zahl  $i \in \{1, \dots, |V|\}$  zugeordnet. Gestartet wird die Traversierung an der Wurzel mit der Initialisierung der Variable  $i = 1$ . Zuerst wird der Unterbaum des linken Kindes traversiert, dann wird der aktuelle Knoten  $v \in V$  besucht und zuletzt der Unterbaum des rechten Kindes. Bei jedem Besuch eines Knotens  $v$  wird dem Knoten  $v$  der aktuelle Wert von  $i$  zugewiesen und  $i$  um eins erhöht. Andere Durchläufe sind beispielsweise die *Preorder-Traversierung*, in der zuerst der Knoten sodann der Unterbaum des linken Kindes und zuletzt der Unterbaum des rechten Kindes besucht wird. In der *Postorder-Traversierung* werden zuerst der Unterbaum des linken Kindes und dann des rechten Kindes traversiert und zuletzt der aktuelle Knoten besucht.

Wie die folgenden Beispiele belegen, werden Baumzeichnungen werden unter anderem in der Informatik, der Mathematik sowie auch im nicht-wissenschaftlichen Umfeld benötigt. Ein Anwendungsgebiet findet sich in der Phylogenetik, in der evolutionäre Beziehungen dargestellt werden. Verschiedene Arten sind zueinander verwandt, was in den Zeichnungen phylogenetischer Bäume dargestellt wird, siehe Abbildung 3.1(a). In der kreisförmigen Darstellung kann man ablesen, welche Arten gemeinsame Vorfahren haben. Auch in der objektorientierten Modellierung werden baumähnliche Strukturen verwendet, da einfache Vererbungshierarchien (ohne Mehrfachvererbung) Bäume darstellen. Im nicht-wissenschaftlichen Umfeld werden Stammbäume in der Ahnenforschung zum Darstellen der Abstammung verwendet. Ein weiteres Beispiel zeigt Abbildung 3.1(b). Darin steht der ausgefüllte Knoten für eine Person  $P$ , deren Beziehungen zu anderen Personen betrachtet werden soll. Es kann einfach erkannt werden, welche Personen direkt mit der Person  $P$  in Beziehung stehen. Des Weiteren lässt sich einfach der Abstand zwischen nicht direkt in Beziehung stehenden Personen ermitteln. Bedingte Wahrscheinlichkeiten werden in der Stochastik mit Baumzeichnungen illustriert. Der Ereignisbaum zeigt mögliche

nacheinander eintretende Ereignisse  $A$  oder  $\bar{A}$  und  $B$  oder  $\bar{B}$ . Dabei tritt erst das Ereignis  $A$  mit Wahrscheinlichkeit  $P(A)$  oder  $\bar{A}$  mit  $P(\bar{A})$  ein. Abhängig vom ersten Ereignis tritt dann  $B$  mit einer Wahrscheinlichkeit  $P_A(B)$  oder  $P_{\bar{A}}(B)$  ein. D. h. die Wahrscheinlichkeit ist abhängig vom Ausgang des ersten Ereignisses ( $A$  oder  $\bar{A}$ ). Um auf einen Blick schnell erfassen zu können, wie die bedingten Wahrscheinlichkeiten zusammenhängen, werden Zeichnungen erstellt, siehe Abbildung 3.1(c).

Übersichtlich dargestellte Zeichnungen ermöglichen eine schnelle Informationsaufnahme und eine effiziente Interpretation. Dies ist notwendig, um davon schnell Entscheidungen abzuleiten oder sich einen Überblick über Begebenheiten in den verschiedenen bereits angeführten Gebieten zu verschaffen.

Wie schon in Abschnitt 2.2 beschrieben, ist es auch beim Zeichnen von Bäumen das Ziel, möglichst „schöne“ und „übersichtliche“ Zeichnungen zu erstellen, damit der Betrachter schnell die Hauptinformationen aus dem Schaubild ziehen kann. Das Zeichnen von Bäumen ist eines der beststudierten Gebieten im Graphenzeichnen. Deshalb gibt es zahlreiche Konventionen und Kriterien, nach denen viele verschiedene Zeichenarten definiert und erforscht wurden.

## 3.1 Bisherige Ansätze

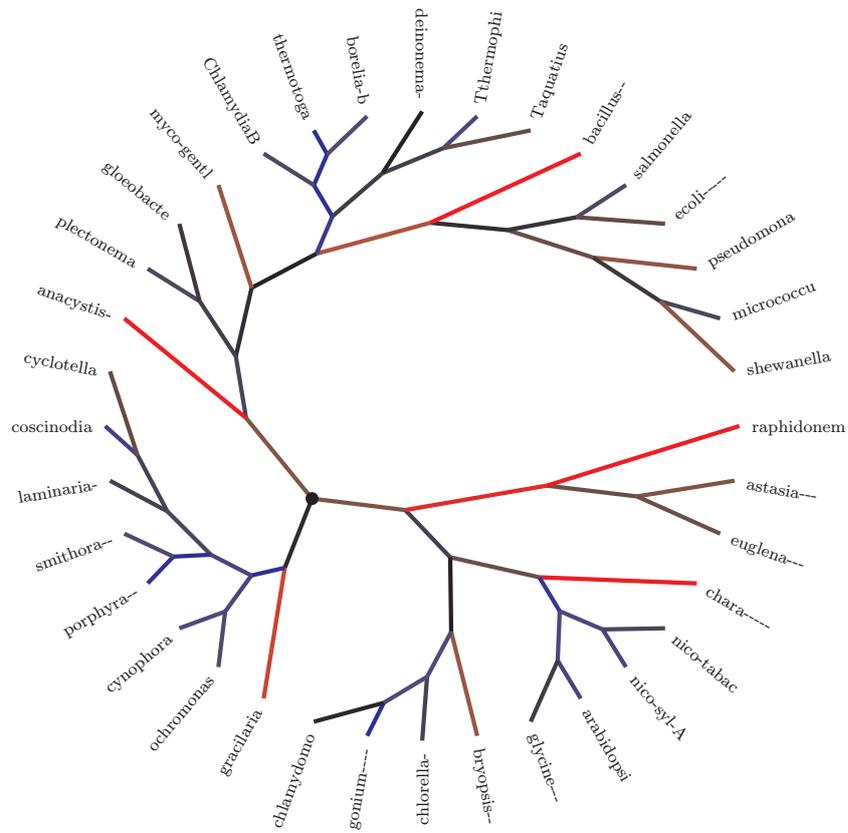
Zu Beginn dieser Forschungsarbeiten vor gut vierzig Jahren stellte Donald E. Knuth die Frage, wie man einen Baum zeichnen solle („How shall we draw a tree?“, [Knu68]). Mit dieser Frage beschäftigten sich daraufhin viele Forscher. Welche Qualitäten soll die Zeichnung eines Baumes besitzen? Wie soll so ein Baum gezeichnet werden? Welches Verfahren kann man dazu verwenden? Wie effizient ist ein Verfahren? Kann es ein effizientes Verfahren geben?

### 3.1.1 Hierarchisch

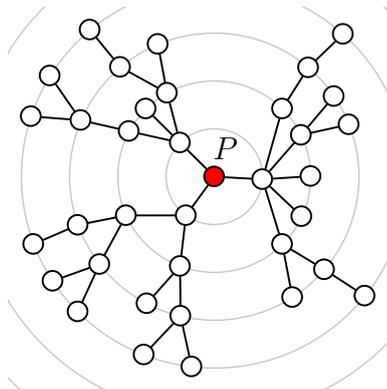
Knuth schlug vor, Binärbäume hierarchisch zu zeichnen: Zeile für Zeile (von oben nach unten) und von links nach rechts, indem man eine Schreibmaschine als Zeichenwerkzeug verwendet [Knu68]. Eine Kante sollte dabei als geradlinige Verbindung zwischen zwei Knoten fungieren, wie es auch schon Fáry 1948 für planare Graphen vorgeschlagen hatte [Fá48]. Eine wichtige Anwendung hierzu ist der binäre Suchbaum, in dem Informationen geordnet gespeichert und schnell gefunden werden können. Zur Darstellung dieser Bäume wurden u. a. Kriterien zum Bewerten der Zeichnungen vorgestellt.

Zu den Zeichenkonventionen ganzzahlige Koordinaten der Knoten, Planarität und das Einhalten eines Mindestabstands kommt das erste Ästhetikkriterium zum hierarchischen Zeichnen von Bäumen:

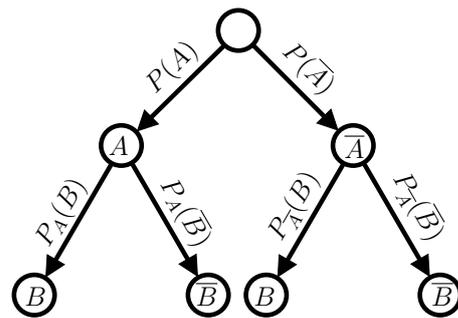
**Definition 3.2 (AK1).** *Knoten auf dem gleichen Level des Baumes sollen auf einer Geraden platziert werden, und die Geraden, die die Level repräsentieren, sollen parallel gezeichnet sein [Knu71, WS79].*



(a) Phylogenetische Bäume [BBS05]

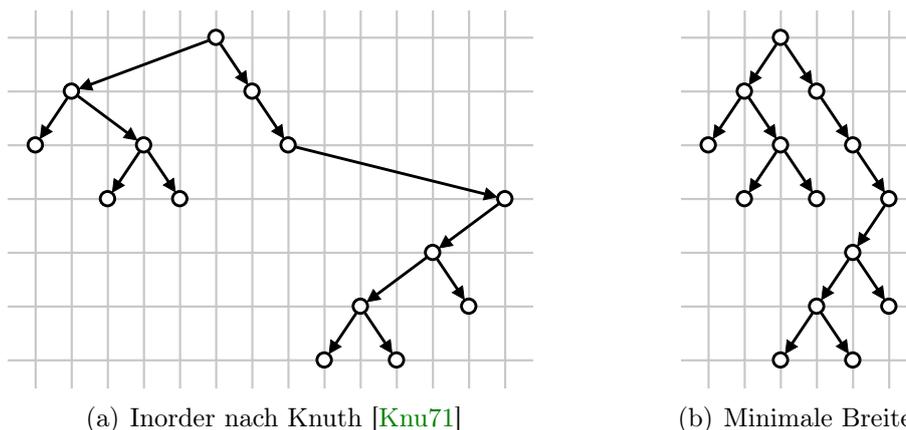


(b) Soziale Netzwerke



(c) Stochastik

Abbildung 3.1. Anwendungsbeispiele für Bäume



**Abbildung 3.2.** Erste Ansätze hierarchischer Baumzeichnungen (aus [WS79])

Angenommen, die Geraden sind horizontal in gleichem Abstand gezeichnet und  $AK1$  wird beachtet, so steht für jeden Knoten die  $y$ -Koordinate bereits fest. Daraus ergeben sich strenge Aufwärtszeichnungen. Die  $x$ -Koordinaten können durch die vergebenen Nummern aus einem Inorderdurchlauf festgelegt werden [Knu71], siehe Abbildung 3.2(a). Wie zu sehen ist, liegen alle Knoten desselben Levels auf einer Geraden und jeder Knoten belegt, entsprechend der Inorder-Traversierung, exklusiv eine Spalte in der Zeichnung. Somit ist die Höhe  $H$  der Zeichnung eines geordneten Binärbaums  $T = (V, E)$  mit der Höhe  $h(T)$  festgelegt. Die Breite ist festgelegt durch die Anzahl der Knoten von  $T$ , da jedem Knoten eine eigene Spalte zugewiesen wird. Im schlechtesten Fall ist die Höhe  $H$  der Zeichnung für eine Liste linear, womit sich eine quadratische Fläche ergibt. Betrachtet man die Zeichnung, so ist sie nicht symmetrisch und bietet viel Platz für Einsparungen in der Breite, siehe Abbildung 3.2(b). Positiv zu bewerten ist jedoch, dass aus der Zeichnung ablesbar ist, welche Knoten linke Kinder sind und welche Knoten rechte Kinder sind, da ein linkes Kind immer links vom Vater gezeichnet ist (und umgekehrt für rechte Kinder). Folglich wird die vorgegebene Ordnung eines Baumes in der Zeichnung beachtet. Solche Zeichnungen werden *ordnungserhaltende Zeichnungen* genannt.

Wetherell und Shannon stellten zunächst ein einfaches Verfahren zum hierarchischen Zeichnen von geordneten gerichteten  $d$ -nären Wurzelbäumen vor, das Bäume mit vorgegebenen Zeichenkonventionen und Ästhetikkriterien auf minimale Fläche zeichnet [WS79].

Als Zeichenkonventionen und Ästhetikkriterien zum Zeichnen der  $d$ -nären Wurzelbäume forderten sie Planarität, kleine Fläche, ganzzahlige Koordinaten und  $AK1$ . Des Weiteren werden Knoten punktförmig und die Kanten als geradlinige Verbindungslinien zwischen adjazenten Knoten repräsentiert. Sie legten die Qualität oder „Schönheit“ einer Zeichnung fest, indem die Zeichnung eine visuelle Struktur besitzen sollte, die die Eigenschaften des vorgegebenen Baumes widerspiegelt. Aufgrund der durch den gegebenen Baum vorherbestimmten Höhe einer Zeichnung, war es zur Mi-

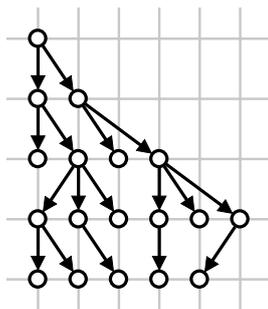


Abbildung 3.3. Linksbündig aus [WS79]

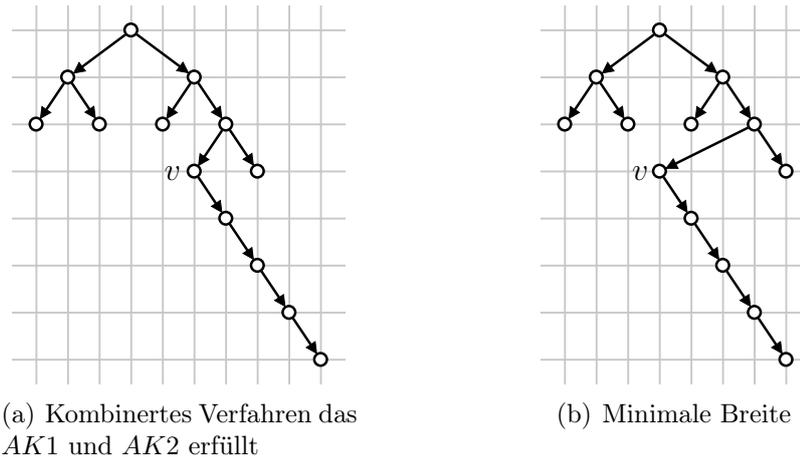
nimierung der Fläche notwendig, die Breite von Zeichnungen zu minimieren. Wetherell und Shannon präsentierten ein iteratives Linearzeit-Zeichenverfahren [WS79], das alle Knoten einer Ebene linksbündig zeichnet, siehe Abbildung 3.3. Die Resultate lieferten zwar minimale Breite unter Beachtung von AK1, aber aufgrund der fehlenden Symmetrie widmeten sie sich einem weiteren Ästhetikkriterium zu Binärbäumen.

**Definition 3.3 (AK2).** *In einem Binärbaum soll ein linkes Kind links vom Vater und ein rechtes Kind rechts vom Vater platziert werden [WS79].*

Dieses Kriterium beachtete bereits das Inorder-Verfahren von Knuth. Das Problem der Zeichnungen, die nach dem Verfahren von Knuth berechnet werden, ist, dass für jeden Knoten eine eigene Spalte reserviert wird, womit die Breite nicht minimal ist, sondern maximal. Das Inorder-Verfahren von Knuth beachtet zwar AK2, aber nicht die minimale Breite. Im Gegensatz dazu, wird in den Zeichnungen nach dem Linksbündig-Verfahren das Ästhetikkriterium AK2 nicht erfüllt, jedoch die minimale Breite. Eine Kombination aus dem Linksbündig- und Inorder-Verfahren liefert zwar Zeichnungen, die AK1 und AK2 erfüllen, aber es können daraus auch sehr breite Zeichnungen resultieren [WS79]. Im Verfahren von Wetherell und Shannon wird zuerst in einem Postorder-Durchlauf jedem Knoten eine vorläufige  $x$ -Koordinate zugewiesen. Zusätzlich werden an den internen Knoten Werte gespeichert, die angeben, um wieviel die Kinder relativ zum Vater nach rechts geschoben werden müssen. In einem zweiten Preorder-Durchlauf werden den Knoten die endgültigen  $x$ -Koordinaten zugewiesen. Diese ergeben sich aus der Summe der vorläufigen  $x$ -Koordinaten und den zusätzlichen relativen Werten für die Unterbäume der Kinder, siehe Abbildung 3.4(a).

Möchte man die Breite verringern, dann würde sich ergeben, dass Zeichnungen zwar schmaler würden, jedoch auch asymmetrischer, wenn man die Kinder hinsichtlich ihres Vaters betrachtet, siehe Abbildung 3.4(b). Deshalb wurde ein weiteres drittes Ästhetikkriterium gefordert.

**Definition 3.4 (AK3).** *Der Vaterknoten soll zentriert über den Kindern platziert werden [WS79].*



**Abbildung 3.4.** AE1, AE2 und kleine Breite (aus [WS79])

Es ist zu beachten, dass sich  $AK3$  zur minimalen Breite konträr verhält, wofür das Beispiel in [Abbildung 3.4](#) dient. Um die Fläche der Zeichnung in [Abbildung 3.4\(a\)](#) zu verringern kann der Knoten  $v$  weiter nach links geschoben werden, siehe [Abbildung 3.4\(b\)](#). Die Folge ist aber, dass  $AK3$  nicht mehr erfüllt wird.

Reingold und Tilford stellten 1981 „den“ Algorithmus zum hierarchischen ordnungserhaltenden Zeichnen von geordneten Binärbäumen vor [RT81]. Sie übernahmen die Ästhetikkriterien ihrer Vorgänger [Knu68, Knu71, WS79], erkannten jedoch Probleme in den Verfahren, weil gleiche Teile von Bäumen nicht gleich gezeichnet wurden.

**Definition 3.5 (Isomorphie).** Seien  $G_1 = (V_1, E_1)$  und  $G_2 = (V_2, E_2)$  zwei Graphen. Die Abbildung  $\phi : V_1 \mapsto V_2$  ist ein Isomorphismus, wenn gilt,

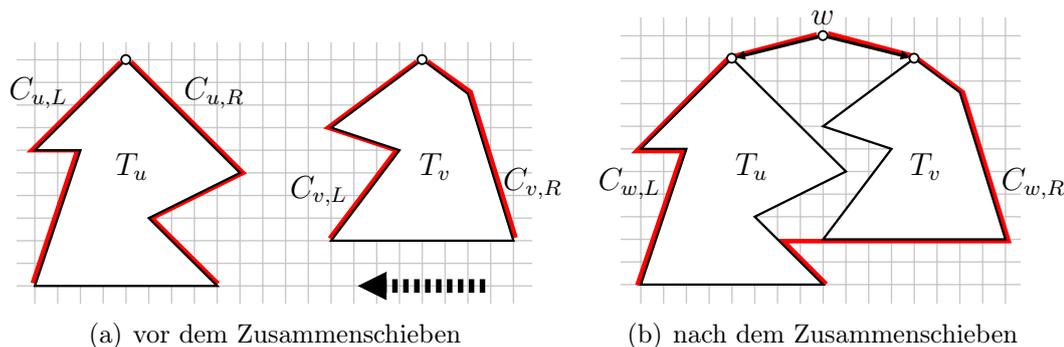
$$(u, v) \in E_1 \Leftrightarrow (\phi(u), \phi(v)) \in E_2,$$

d. h. es gibt eine bijektive Abbildung zwischen  $G_1$  und  $G_2$ .

Deshalb stellten sie die folgende Forderung, um gleiche Strukturen in einer Zeichnung bei Betrachtung der Zeichnung schneller erkennen zu können.

**Definition 3.6 (AK4).** Ein Baum und sein Spiegelbild sollen Zeichnungen produzieren, die Spiegelungen voneinander sind, und isomorphe Unterbäume sollen bis auf Verschiebung gleich gezeichnet werden [RT81].

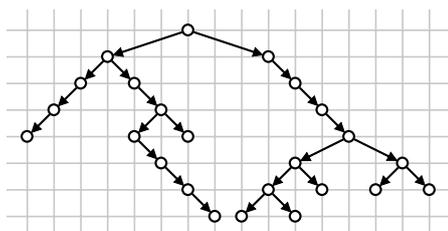
Es ist anzumerken, dass das Inorder-Verfahren von Knuth dieses Ästhetikkriterium erfüllt. Bis heute wird es als ein wichtiges Kriterium angesehen, da so ähnliche Strukturen in Zeichnungen leichter erkannt werden können. Das dazu von Reingold und Tilford entwickelte Divide & Conquer Verfahren beinhaltet zwei Durchläufe,



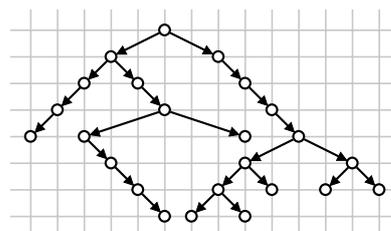
**Abbildung 3.5.** Rekursionsschritt Reingold und Tilford Algorithmus

einen ersten Postorder- und einen zweiten Preorder-Durchlauf. Im ersten Durchlauf werden den Knoten  $u$  und  $v$  relativ zum Vaterknoten  $w$   $x$ -Koordinaten zugewiesen. Anschaulich wird der Unterbaum  $T_v$  von rechts so weit an  $T_u$  geschoben, dass der Mindestabstand erhalten bleibt, siehe Abbildung 3.5(a). Danach wird die  $x$ -Koordinate des Vaters zentriert über den beiden nebeneinander platzierten Unterbäumen gezeichnet, siehe Abbildung 3.5(b). Ist Ganzzahligkeit gefordert, so muss gegebenenfalls der rechte Unterbaum wieder um eins nach rechts verschoben werden, damit die  $x$ -Koordinate des Vaters ganzzahlig bleibt und gleichzeitig *AK3* erfüllt ist. Wenn ein Knoten keine Kinder hat, dann ist seine Zeichnung trivialerweise festgelegt.

Um eine effiziente Implementierung zu garantieren, führen Reingold und Tilford sogenannte *Konturen* (engl. threads) ein. Die linke (rechte) Kontur  $C_{v,L}$  ( $C_{v,R}$ ) einer Zeichnung  $\Gamma(T_v)$  eines Unterbaums  $T_v$  mit Wurzel  $v$  ist die verkettete Liste aller linken (rechten) Knoten der Level  $1, \dots, h(T_v)$  in  $\Gamma(T_v)$ . Der erste Knoten von  $C_{v,L}$  bzw.  $C_{v,R}$  ist dabei immer die Wurzel  $v$  des Unterbaums. In Abbildung 3.5(a) sind zwei bereits gezeichnete Unterbäume  $T_u$  und  $T_v$  mit ihren Konturen  $C_{u,L}$ ,  $C_{u,R}$ ,  $C_{v,L}$  und  $C_{v,R}$  skizziert. Um den minimalen horizontalen Abstand zwischen  $T_u$  und  $T_v$  zu bestimmen, werden die einander zugewandten Konturen  $C_{u,R}$  und  $C_{v,L}$  verwendet. Es wird simultan ebenenweise bis zum Ende der kürzeren Kontur der minimale Abstand bestimmt. Das Minimum dieser Abstände minus eins gibt an, um wieviel  $T_v$  nach links geschoben werden kann, ohne dabei die Planarität zu zerstören. Als nächstes wird der Vaterknoten  $w$  von  $u$  und  $v$  zentriert über die aktuellen Positionen von  $u$  und  $v$  gesetzt, siehe Abbildung 3.5(b). Falls der Mittelwert der  $x$ -Koordinaten von  $u$  und  $v$  keine ganze Zahl ist, wird  $T_v$  um eins nach rechts geschoben, um dann  $w$  auf eine ganzzahlige Koordinate setzen zu können und um *AK3* zu erfüllen. Die Kontur  $C_{w,L}$  ( $C_{w,R}$ ) des aktuellen Unterbaums  $T_w$  von  $w$  ergibt sich aus der Konkatenation  $w \circ C_{u,L}$  ( $w \circ C_{v,R}$ ). Ist  $T_u$  höher als  $T_v$ , also  $h(T_u) > h(T_v)$ , dann ist die rechte Kontur  $C_{w,R}$  die Konkatenation von  $C_{v,R}$  mit der Teilliste ab der Höhe  $h(T_v) + 1$  von  $C_{u,R}$ , siehe Abbildung 3.5(b). Wenn  $T_v$  höher ist als  $T_u$ , so wird die linke Kontur  $C_{w,L}$  entsprechend erweitert. Sind beide Unterbäume  $T_u$  und  $T_v$  gleich



(a) Zeichnung mit Reingold und Tilford Algorithmus mit Breite 15



(b) Schmälere Zeichnung als in Abbildung 3.6(a) mit Breite 13

**Abbildung 3.6.** Beispielzeichnung eines Binärbaumes

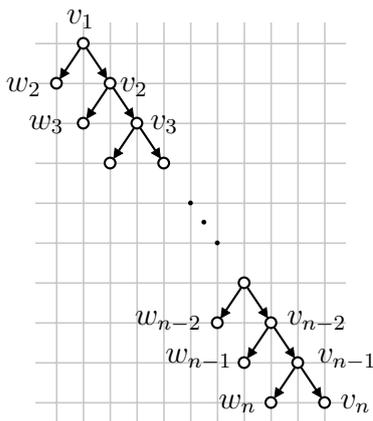
hoch, so entfällt dieser Schritt.

Werden bei der Distanzberechnung in jedem Rekursionsschritt die Konturen  $C_{u,R}$  und  $C_{v,L}$  immer nur bis zum Ende der kürzeren Kontur abgelaufen, wird eine Laufzeit in  $\mathcal{O}(n)$  erreicht [RT81].

Dieses Zeichenverfahren wurde zur gängigsten Art Binärbäume hierarchisch zu zeichnen, da Ästhetikkriterien Ganzzahligkeit, Planarität, geradlinige Kanten, AK1, AK2, AK3 und die identischen Zeichnungen isomorpher Unterbäume (bis auf Verschiebung und Spiegelung) beachtet. Reingold und Tilford machten sozusagen den Ansatz für Binärbäume mit ihrem Verfahren praxistauglich. Das Ziel der minimalen Breite wird jedoch mit den gegebenen Ästhetikkriterien nicht erreicht, was in Abbildung 3.6 gezeigt wird. Da der Algorithmus in jedem Rekursionsschritt versucht nur lokal die minimale Breite zu erreichen, kommt es in diesem Baum nicht zur minimalen Breite. In Abbildung 3.6(a) ist eine Zeichnung gezeigt, die das Verfahren berechnen würde. In Abbildung 3.6(b) ist eine Zeichnung zu sehen, die um zwei schmaler ist und trotzdem die Ästhetikkriterien beachtet.

Dem Problem der minimalen Breite von ganzzahligen hierarchischen Zeichnungen von geordneten Binärbäumen unter Beachtung der gegebenen Ästhetikkriterien haben sich 1982 Supowit und Reingold gewidmet und mit einer Reduktion von  $\beta$ -SAT bewiesen, dass das Bestimmen der minimalen Breite  $\mathcal{NP}$ -hart ist [SR82, ABJT04]. Wird die Ganzzahligkeit außer Acht gelassen, so kann mittels Linearer Programmierung die minimale Breite in Polynomzeit berechnet werden. Für ungeordnete Binärbäume ist das Bestimmen der minimalen Breite einer nicht-ordnungserhaltenden hierarchischen Zeichnung ebenfalls  $\mathcal{NP}$ -hart, was mit einer Reduktion von SAT von Marriott und Stuckey bewiesen wurde [MS04]. Als Zeichenkonventionen und Ästhetikkriterien werden das ebenenweise Zeichnen, Planarität, Ganzzahligkeit, Geradlinigkeit, horizontaler Minimalabstand von 1 und Vater zentriert über den Kindern gefordert.

Die untere Schranke für die benötigte Fläche für ordnungserhaltende hierarchische Zeichnungen mit den Ästhetikkriterien AK1 bis AK4 ist in  $\Omega(n^2)$ . Dazu genügt das Beispiel in Abbildung 3.7, in dem jeder Knoten  $v_i$  mit  $i \in \{1, \dots, n-1\}$  und  $n \in \mathbb{N}$  ein rechtes Kind  $v_{i+1}$  besitzt und ein linkes Kind  $w_{i+1}$ . Die Höhe dieses Bau-



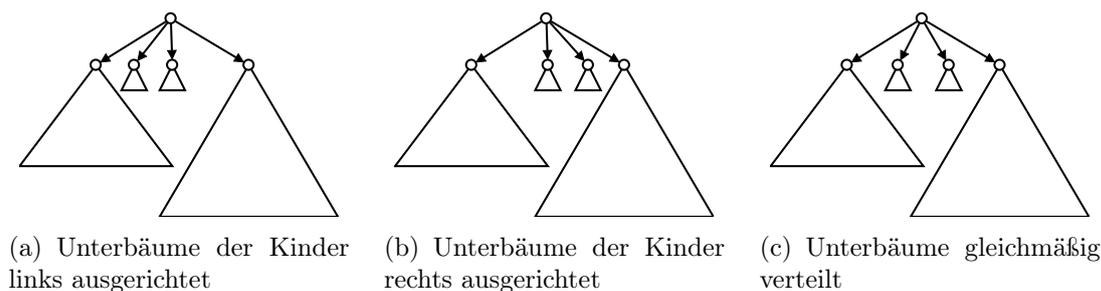
**Abbildung 3.7.** Untere Schranke in  $\Omega(n^2)$  mit Ästhetikkriterien  $AK1$  bis  $AK4$

mes ergibt sich aus  $AK1$  in  $\Omega(n)$ . Da sich die Breite aufgrund von  $AK2$  für jedes Level um eins erhöht, ist die Breite in  $\Omega(n)$ , womit die Fläche in  $\Omega(n^2)$  ist [RT81].

Eine Erweiterung des Algorithmus von Reingold und Tilford für Bäume von beliebigem Ausgangsgrad präsentiert Walker [Wal90]. Dazu werden die Ästhetikkriterien Ganzzahligkeit und  $AK1$  bis  $AK4$  vom binären Fall angepasst.  $AK1$  kann problemlos übernommen werden. Aber das Ästhetikkriterium  $AK2$ , „linkes Kind links vom Vater“ und „rechtes Kind rechts vom Vater“ kann bei mehreren Kindern in Zeichnungen von  $d$ -nären Bäumen nicht analog übernommen werden. Somit wird  $AK2$  im Folgenden so interpretiert, dass die links-rechts Reihenfolge der Kinder in der Zeichnung beachtet werden soll. Das Kriterium  $AK3$  hingegen, die Positionierung des Vaters zentriert über den Kindern, wird dahingehend übernommen, dass das linke und das rechte Kind diesbezüglich betrachtet werden und somit der Vater zentriert über diesen platziert ist. Das Ästhetikkriterium  $AK4$  zu identischen Zeichnungen von isomorphen Unterbäumen und spiegelverkehrten Bäumen kann jedoch ohne Anpassung übernommen werden.

Anschaulich betrachtet, arbeitet das Verfahren ähnlich zum Algorithmus von Reingold und Tilford. Es werden die Unterbäume der Kinder eines Knotens im Rekursionsschritt von rechts soweit wie möglich an die sich links befindenden Unterbäume geschoben. Mit diesem Vorgehen kann es jedoch passieren, dass Asymmetrien auftreten, siehe Abbildung 3.8. In Abbildung 3.8(a) werden die Unterbäume von rechts an die links davon liegenden Unterbäume geschoben und in Abbildung 3.8(b) von links nach rechts. Somit wird das Spiegelbild von geordneten  $d$ -nären Bäumen nicht gleich gezeichnet, was  $AK4$  verletzt. Die Lösung präsentiert Walker, indem er kleine Kindunterbäume gleichmäßig zwischen großen Kindunterbäumen verteilt, siehe Abbildung 3.8(c). Diese Asymmetrien werden somit durch das von Walker entwickelte Verfahren behoben [Wal90], das jedoch ein nicht mehr ganzzahliges Ergebnis zur Folge hat.

Walker behauptet, dass die Laufzeit seiner Implementierung für das präsentierte

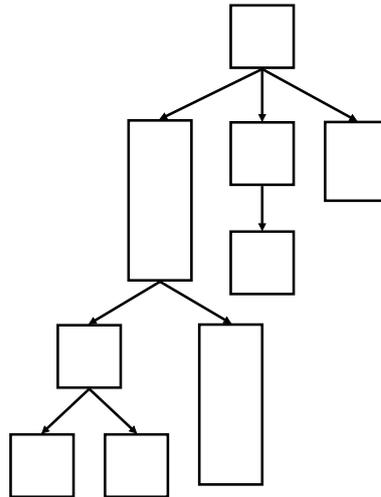


**Abbildung 3.8.** Problem der Asymmetrie für  $d$ -näre Bäume [Wal90]

Verfahren in  $\mathcal{O}(n)$  ist [Wal90]. Buchheim et al. korrigierten das Verfahren geringfügig, um damit die gewünschte lineare Laufzeit zu erreichen [BJL02]. Somit werden gewurzelte  $d$ -näre Bäume hierarchisch unter Beachtung der Ästhetikkriterien Ganzzahligkeit, Knoten auf Level (*AK1*), Links-Rechts-Ordnung (*AK2*), Vater zentriert über den Kindern (*AK3*), identische Zeichnungen für isomorphe Unterbäume, deren Spiegelbild (*AK4*) in Linearzeit sowie in quadratischer Fläche ganzzahlig gezeichnet.

Als weitere Ästhetikkriterien, die zwar nicht gefordert sind, aber trotzdem betrachtet werden können, gibt es: Maximale Kantenlänge, Uniformität der Kantenlängen, Seitenverhältnis oder Winkelauflösung, siehe Abschnitt 2.2.2. Da die vertikalen Abstände zwischen Vaterknoten und den Kindern durch die Abstände der aufeinanderfolgenden Levels festgelegt sind, ist für die Kantenlänge nur noch der Abstand in horizontaler Richtung relevant. Nimmt man beispielsweise einen vollständigen gewurzelten Binärbaum (oder auch gewurzelten  $d$ -nären Baum), dann haben die ausgehenden Kanten der Wurzel eine Länge in  $\Omega(n)$ , da die Breite durch die Anzahl der Blätter auf der untersten Ebene vorgegeben und die Kinder mittig über ihren Unterbäumen platziert sind. Damit ergibt sich auch eine Varianz der Kantenlängen in  $\Omega(n)$ . Das Seitenverhältnis eines hierarchisch gezeichneten Baumes kann ebenfalls in  $\frac{H}{W} \in \Omega(n)$  sein. Man betrachte dazu den Baum aus Abbildung 3.7 und vertausche für  $i \bmod 2 = 0$  die Kinder von  $v_i$ . Dann ist die Höhe der Zeichnung in  $\Omega(n)$  und die Breite konstant ist. Dadurch ergibt sich eine untere Schranke für das Seitenverhältnis in  $\Omega(n)$ . Das Kriterium der Winkelauflösung kann für  $d$ -näre Bäume ebenfalls schlecht erfüllt werden, wenn ein Knoten sehr viele Kinder hat. Dann werden die Winkel sehr klein und die Unterscheidung der einzelnen ausgehenden Kanten wird erschwert.

Da es oft notwendig ist, die Knoten zu beschriften wurde von Bloesch ein Ansatz für rechteckig gezeichnete Knoten mit unterschiedlicher Größe vorgestellt [Blo93]. Innerhalb dieser Rechtecke können die Beschriftungen angebracht werden. Nützlich sind diese Zeichnungen in der Darstellung von Klassenhierarchien in der objektorientierten Modellierung. Hierzu liefern Ansätze von Wetherell und Shannon, Reingold und Tilford und Walker oft nicht adäquate Zeichnungen. Die oben präsentierten Zeichenkonventionen und Ästhetikkriterien sollen hier ebenfalls beachtet werden. Dazu



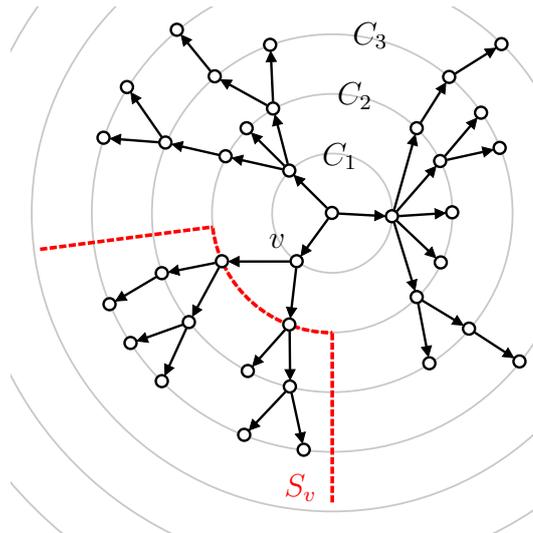
**Abbildung 3.9.** Hierarchische Baumzeichnung mit rechteckigen Knoten [Blo93]

ergeben sich aufgrund der rechteckigen unterschiedlich großen Knoten jedoch noch weitere Kriterien. Geschwisterknoten sollen am oberen Rand an derselben Horizontalen ausgerichtet sein und es soll eine feste vertikale Distanz zwischen Vater- und dessen Kindknoten eingehalten werden. Eine Beispielzeichnung eines solchen Baumes ist in [Abbildung 3.9](#) gezeigt.

In Baumzeichnungen ist es oft notwendig, knotendisjunkte Unterbäume in Zeichnungen leicht voneinander unterscheiden zu können. Seien  $T_v$  und  $T_w$  mit den Wurzeln  $v$  und  $w$  diskunkte Unterbäume eines Baumes  $T$ . Sind in einer Zeichnung die *rechteckigen konvexen Hüllen* von  $T_v$  und  $T_w$  disjunkt, dann gilt die Eigenschaft der *rechteckigen Unterbaum Separation* als erfüllt [CGKT02]. Es ist zu beachten, dass die Algorithmen von Reingold und Tilford und auch von Walker diese Art von Unterbaum Separation nicht erfüllen, da Unterbäume sich in deren resultierenden Zeichnungen ineinander verzahnen können.

### 3.1.2 Radial

Zu den bereits beschriebenen hierarchischen Ansätzen in [Abschnitt 3.1.1](#) führt Eades radiale Zeichnungen von Bäumen mit punktförmigen Knoten ein [Ead92]. Bäume werden hier zentralisiert dargestellt mit der Wurzel in der Mitte. Die Knoten mit der Tiefe  $d$  liegen dabei auf dem  $d$ -ten konzentrischen Ring  $C_d$ , wobei für die entsprechenden Radien  $r_i < r_{i+1}$  gilt. Die Kanten verlaufen vom  $i$ -ten und zum  $(i + 1)$ -ten Ring, siehe [Abbildung 3.10](#). Alle Knoten halten einen bestimmten Mindestabstand ein. Dieser Ansatz wird in sozialen Netzwerken verwendet. Ein Knoten, der zu einer bestimmten Person gehört, wird in das Zentrum platziert. Alle benachbarten Knoten (Personen), die mit der Person direkt in Verbindung stehen, werden auf den ersten konzentrischen Ring gezeichnet. Die Freundesfreunde dann auf den zweiten



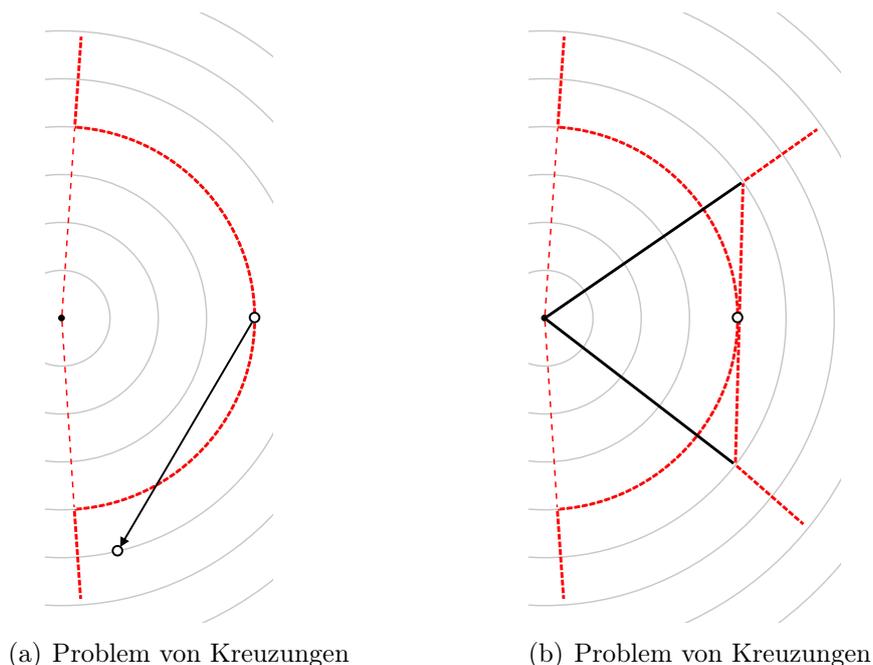
**Abbildung 3.10.** Radialer Ansatz für  $d$ -näre Bäume (aus [DBETT99, Ead92])

Ring, und so fort. Mit dieser Zeichenart ist die wichtigste Person oder die aktuelle Hauptperson in der Mitte platziert. Der Abstand kann in solchen Zeichnungen von ausgewählten Personen zur Hauptperson an der Nummer des entsprechenden konzentrischen Rings abgelesen werden. Im Gegensatz zu hierarchischen Zeichnungen werden im radialen Fall die Knoten auf reelle Koordinaten platziert und Kanten können im radialen Fall in alle Richtungen zeigen, d. h. die Zeichnungen sind keine Aufwärtszeichnungen mehr, wie es in hierarchischen Zeichnungen der Fall war.

Betrachtet man den Unterbaum  $T_v$  eines Knotens  $v$ , so wird  $T_v$  in einem Sektor  $S_v$  gezeichnet. Der Winkel, der von  $S_v$  eingeschlossen wird, ist proportional zur Anzahl der Blätter im Unterbaum von  $T_v$ . Mit dieser Methode können aber Kanten-Kreuzungen entstehen, weil dadurch Kanten den Sektor verlassen können, siehe Abbildung 3.11(a). Um dieses Problem in den Griff zu bekommen, werden die Knoten auf einen konvexen Teil von  $S_v$  platziert, siehe Abbildung 3.11(b).

Betrachtet man diese Zeichnungen aus Sicht der in Abschnitt 3.1.1 beschriebenen Ästhetikkriterien, so ist das Ästhetikkriterium Knoten desselben Levels auf einer Geraden (AK1) hier so verwirklicht, dass die bis zu  $d^k$  Knoten des  $k$ -ten Levels eines  $d$ -nären Baumes auf derselben Kreislinie desselben  $k$ -ten konzentrischen Kreises platziert sind. Die Eigenschaft der Rechteck Unterbaum Separation wird in radialen Layouts so erfüllt, dass disjunkte Unterbäume in disjunkten Sektoren liegen. Somit können verschiedene Teile der Baumzeichnung vom Auge schneller unterschieden werden.

Anwendungen des radialen Layouts finden sich auch im Zeichnen von Peer-to-Peer Netzwerken [YF01]. Es wird ein System beschrieben, womit in einem Netzwerk ein Host markiert werden kann, der dann ins Zentrum gesetzt wird, und der Rest radial gezeichnet wird, siehe Abbildung 3.12. Die konzentrischen Kreise repräsentieren die Distanz im Netzwerk von der Wurzel, dem Host, zu allen anderen Hosts. Ein



**Abbildung 3.11.** Radialer Ansatz für  $d$ -näre Bäume (aus [DBETT99, Ead92])

Knoten mit dem Abstand  $i$  zum markierten Host ist auf dem  $i$ -ten konzentrischen Kreis platziert.

Im Folgenden werden Ästhetikkriterien zu radialen Baumzeichnungen betrachtet. Die Länge der Kanten ist abhängig von den gewählten Radien zwischen den konzentrischen Kreisen. Abhängig von einem gewählten Mindestabstand zwischen Knoten auf dem  $i$ -ten Level kann der Radius von  $C_i$  bestimmt werden. Des Weiteren können die ausgehenden Kanten eines Knotens  $v$  sehr unterschiedliche Längen haben, vor allem, wenn  $v$  viele Kinder besitzt. Somit ist die Uniformität der Kantenlängen in diesem Zeichenansatz nicht gegeben. Betrachtet man  $AK1$ , das ebenenweise Zeichnen, so kann das Ästhetikkriterium dazu adaptiert werden. Alle Knoten desselben Levels werden auf derselben konzentrischen Kreislinie platziert. Somit ist das adaptierte Ästhetikkriterium  $AK1$  sowie wie das Kriterium Planarität erfüllt.

Ähnlich zu den hierarchischen Zeichnungen in Abschnitt 3.1.1 ist eine hohe Winkelauflösung in radialen Zeichnungen nicht sichergestellt, da es Knoten mit sehr vielen Kindern geben kann, so dass die Winkel gezwungenermaßen klein werden. Die untere Schranke für die Fläche von radialen Zeichnungen ist in  $\Omega(n^2)$ , da es vier Pfade von der Wurzel ausgehend geben kann, die eine lineare Höhe und eine lineare Breite der Zeichnung erzwingen.

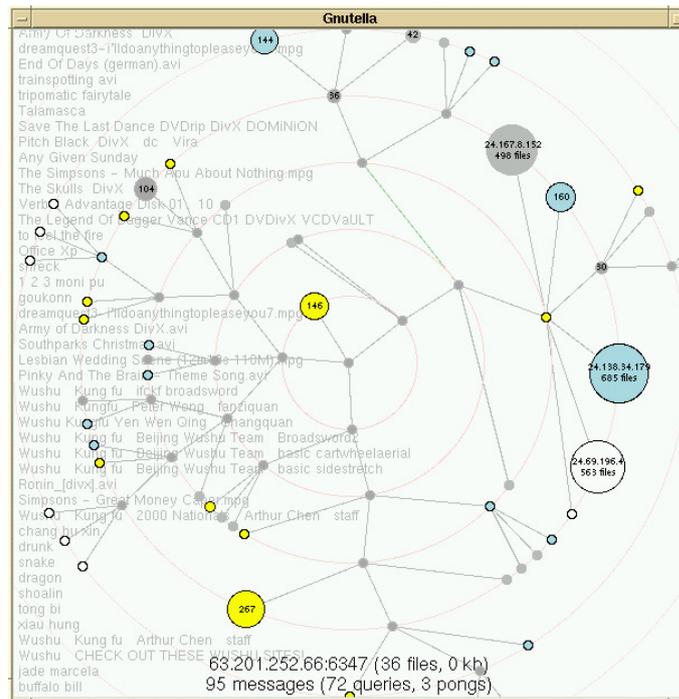


Abbildung 3.12. Radiale Visualisierung des Gnutella Netzwerks [YF01]

### 3.1.3 Gitter

In diesem Abschnitt wird auf die in der Literatur vorhandenen Ansätze zum Zeichnen von Bäumen auf Gittern eingegangen. Dazu werden verschiedene Zeichenkonventionen und Ästhetikkriterien betrachtet, anhand derer die Zeichnungen bewertet werden. Wichtigste Eigenschaft ist, dass die Knoten auf den ganzzahligen Koordinaten oder genauer gesagt, auf den Knoten des Gittergraphens eingebettet sind. Im Folgenden werden nur die Kantenverläufe unterschieden. In erster Linie wurden Zeichnungen von Binärbäumen auf dem 4-er Gitter untersucht. Unter Berücksichtigung der Tatsache, dass Gittereinbettungen auf dem 4-er Gitter Bäume bis zum Grad vier erlauben, wurden Binärbäume, aber auch Ternärbäume untersucht. Ästhetikkriterien waren dazu unter anderem die von der Zeichnung eingenommene Fläche, die maximale Kantenlänge, die Anzahl der Knicke und das Seitenverhältnis von Zeichnungen. Die Kriterien Planarität und ganzzahlige Koordinaten für die Knoten und für eventuelle Kantenknicke gelten im Folgenden automatisch.

Wie auch schon im hierarchischen Fall gibt es viele Arbeiten zum Zeichnen von Binärbäumen auf Gittern, speziell zum orthogonalen oder 4-er Gitter. Die Zeichnungen unterscheiden sich in den diversen Kriterien, die erfüllt werden sollen. Meist werden untere und/oder obere Schranken für die benötigte Fläche unter Berücksichtigung bestimmter Zeichenkonventionen und Ästhetikkriterien diskutiert. In den einzelnen Fällen wird zwischen strengen Aufwärtszeichnungen, Aufwärtszeichnungen

	streng aufwärts	aufwärts	—
geradlinig, orthogonal	$\Theta(n \log n)$ [CDBP92]	$\Theta(n \log n)$ [CDBP92, CGKT02]	$\Theta(n \log \log n)$ [CGKT02, SKC00]
orthogonal		$\Theta(n \log \log n)$ [Kim95, GGT96, SKC00]	$\Theta(n \log \log n)$
polyline		$\Theta(n)$ [GGT96]	$\Theta(n)$ [Lei80, Val81]

**Tabelle 3.1.** Flächenschranken für Binärbäume

und Zeichnungen ohne Einschränkung für die Kantenrichtungen unterschieden.

Um eine Richtung in Zeichnungen zu repräsentieren, z. B. von oben nach unten, wurden Aufwärtszeichnungen und strenge Aufwärtszeichnungen untersucht. In Abbildungen 3.13(i), 3.13(c), 3.13(f) und 3.13(l) haben die Kanten beliebige Richtungen in alle Himmelsrichtungen. Im Vergleich dazu verlaufen die Kanten in Abbildungen 3.13(h), 3.13(b), 3.13(e) und 3.13(k) monoton nach unten. Zuletzt sind in Abbildungen 3.13(g), 3.13(a), 3.13(d) und 3.13(j) strenge Aufwärtszeichnungen gezeigt, wo für jede Kante  $(u, v)$  gelten muss, dass  $v$  unterhalb von  $u$  liegen muss. Zu beachten ist hierbei, dass in Abbildungen 3.13(a) und 3.13(d) waagrechte Kanten bzw. Kantensegmente vorhanden sind, aber die Kanten streng monoton steigend z. B. nach rechts unten verlaufen, so dass diese Zeichnungen trotzdem als strenge Aufwärtszeichnungen bezeichnet werden. Verlaufen die Kanten in einer Gittereinbettung in das 4-er Gitter entweder horizontal nach rechts oder vertikal nach unten, so wird die Einbettung bzw. Zeichnung *hv-Zeichnung* genannt [CDBP92, ELL92].

### 3.1.3.1 Geradlinig und orthogonal

Zuerst werden strenge Aufwärtszeichnungen bzw. *hv-Zeichnungen* mit geradlinig orthogonalen Kanten betrachtet. Aufgrund der höchstens zwei möglichen Richtungen in *hv-Zeichnungen* können nur Bäume vom Grad zwei gezeichnet werden. Die untere und obere Schranke für die benötigte Zeichenfläche für vollständige Binärbäume ist in  $\Theta(n)$  [CDBP92]. Somit ist diese Schranke für Aufwärtszeichnungen und in alle vier Richtungen ebenfalls in  $\Theta(n)$  [CDBP92, Shi76]. Wenn in einer Zeichnung eines Binärbaums die Kanten zu den Kindern immer senkrecht zur eingehenden Kante gezeichnet werden, so wird eine Zeichnung *H-Baum Zeichnung* genannt. H-Baum Zeichnungen von vollständigen Binärbäumen können in  $\mathcal{O}(n)$  Fläche erstellt werden [Lei80, BK80].

Für beliebige ungeordnete Binärbäume können *hv-Zeichnungen* erstellt werden, die  $\Theta(n \log n)$  Fläche mit Seitenverhältnis in  $\mathcal{O}(\frac{n}{\log n})$  [CDBP92] benötigen. Eades

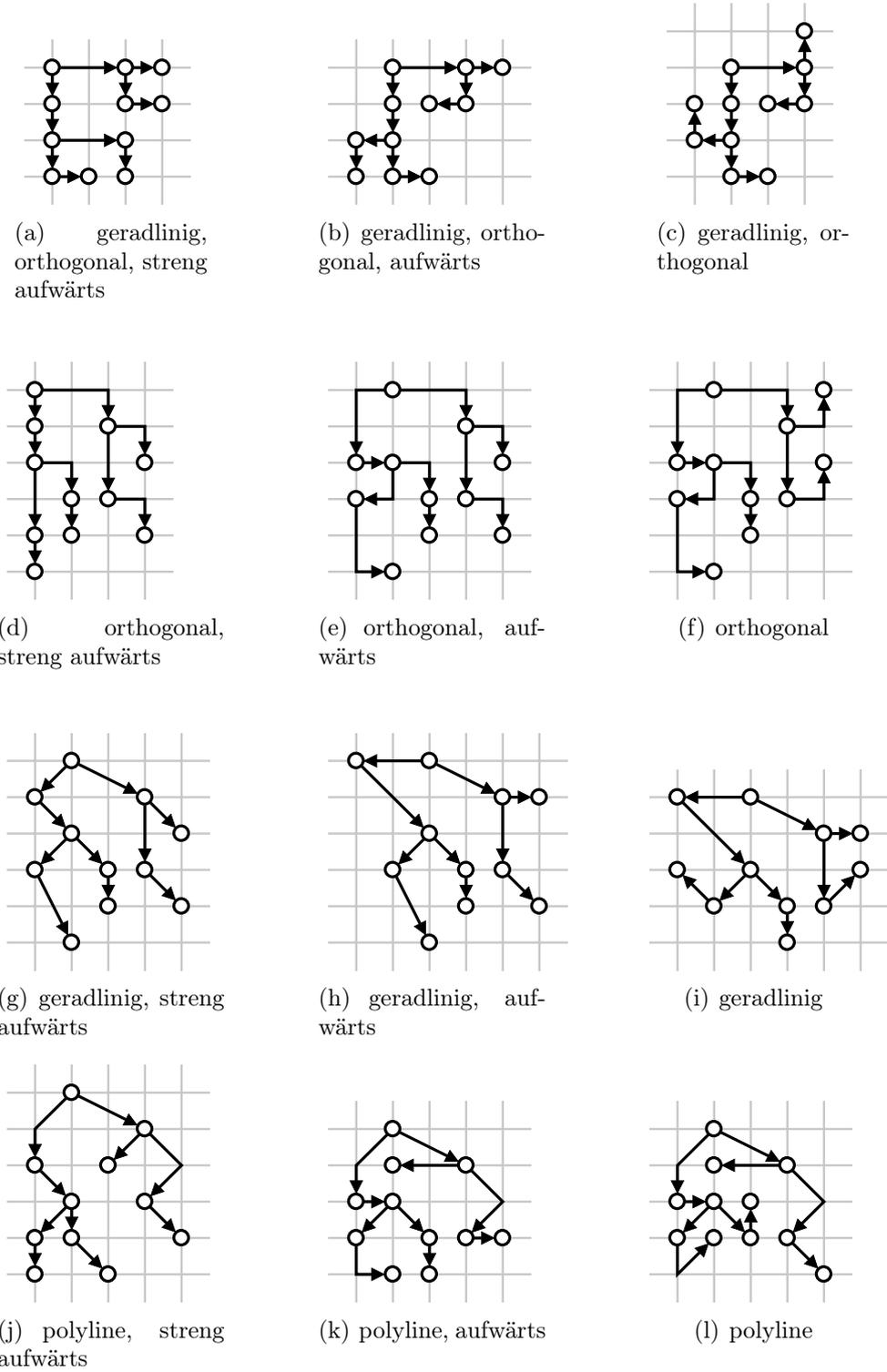


Abbildung 3.13. Gitterzeichnungen

et al. haben ein Verfahren zum Erstellen von  $hv$ -Zeichnungen mit minimaler Fläche bezüglich der rechteckigen Unterbaum Separation vorgestellt, das die Zeichnungen in quadratischer Zeit berechnet [ELL92].

Chan et al. haben für Aufwärtszeichnungen mit einem „recursive winding“ Ansatz die Schranke  $\mathcal{O}(n \log n)$  bestätigt, aber eine Verbesserung bzgl. des Seitenverhältnisses erreicht [CGKT97, CGKT02]. Das Verfahren kann Zeichnungen mit einem guten Seitenverhältnis erzeugen. Mit diesem Ansatz konnten sie ebenfalls eine obere Schranke in  $\mathcal{O}(n \log \log n)$  für geradlinige Gitterzeichnungen in alle Richtungen zeigen. Das Zeichnen von Ternärbäumen [BC87] und Binärbäumen [Gre89] mit einheitlicher Kantenlänge (alle Kanten haben die gleiche Länge) ist aufgrund einer Reduktion von  $\beta$ -SAT  $\mathcal{NP}$ -hart.

Für ordnungserhaltende geradlinige orthogonale Zeichnungen von Binärbäumen ist eine quadratische obere und untere Schranke für (strenge) Aufwärtszeichnungen bekannt [CDBP92, Fra07]. Geordnete Binärbäume können in  $\mathcal{O}(n^{1.5})$  Fläche gezeichnet werden, wenn alle vier Richtungen für die Kanten erlaubt sind [Fra07].

Für vollständige Ternärbäume hat Frati gezeigt, dass diese mit vier verschiedenen Richtungen auf dem 4-er Gitter in  $\mathcal{O}(n^{1.262})$  Fläche gezeichnet werden können [Fra07]. Des Weiteren zeigt er, dass ordnungserhaltende Zeichnungen von Ternärbäumen eine untere und obere Flächenschranke in  $\Theta(n^2)$  haben. Für nicht ordnungserhaltende Zeichnungen präsentiert er ein induktives Verfahren, das Ternärbäume in der Fläche  $\mathcal{O}(n^{1.631})$  zeichnet.

### 3.1.3.2 Orthogonal

Die Kanten von orthogonalen Zeichnungen verlaufen ebenfalls nur auf dem orthogonalen Gitter, jedoch mit möglichen Knicken, die auf ganzzahligen Koordinaten liegen. Dazu wurde gezeigt, dass es möglich ist jeden Baum in  $\mathcal{O}(n)$  Fläche planar und orthogonal in ein Gitter einzubetten, wobei Kantenknicke erlaubt sind [Val81] und die maximale Kantenlänge in  $\mathcal{O}(\sqrt{n})$  ist [RS81]. In das Verkürzen der Kantenlängen in Zeichnungen wurde noch mehr Forschung investiert, da kurze Leitungen auf Chips geringere Materialkosten zur Folge haben [Val81, PRS81, RS81, BL83, Lei84]. Nach den in der Literatur präsentierten Ansätzen ist die Anzahl der Kantenknicke pro Kante in  $\mathcal{O}(\log n)$  und insgesamt in  $\mathcal{O}(n)$ . Diese Schranken werden von Shin et al. verbessert, indem sie ein Verfahren angeben, das Zeichnungen mit Seitenverhältnis in  $[\mathcal{O}(\log^2 n/n), \mathcal{O}(n/\log^2 n)]$  erzeugt. Die Anzahl der Knicke pro Kante ist in  $\mathcal{O}(\log \log n)$  und die Gesamtzahl der Knicke in  $\mathcal{O}(n)$ . Wenn das Seitenverhältnis entweder  $\mathcal{O}(\log^2 n/n)$  oder  $\mathcal{O}(n/\log^2 n)$  ist, so kann die Anzahl der Knicke pro Kante auf  $\mathcal{O}(1)$  reduziert werden. Dies erreichen Shin et al., indem die Fläche leicht erhöht wird, aber trotzdem fast noch in  $\mathcal{O}(n)$  bleibt [SKKC98].

Oft ist es im Chip-Design notwendig, dass die Blätter am Rand des Chips platziert sind. Diese Tatsache haben Brent und Kung aufgegriffen und festgestellt, dass eine orthogonale Zeichnung mit diesen Eigenschaften  $\Omega(n \log n)$  Fläche benötigt [BK80]. Diese Zeichnungen stellen Aufwärtszeichnungen dar, wobei die Hälfte der Kanten genau einen Knick besitzen. Auf dem orthogonalen Gitter wurde für

vollständige Binärbäume von Paterson et al. gezeigt, dass die untere Schranke für die maximale Kantenlänge in Zeichnungen in  $\Omega(\sqrt{n}/\log n)$  ist [PRS81, RS81], was die Ergebnisse von Bhatt und Leiserson [BL83] verbessert.

Garg et al. präsentieren ein Linearzeit-Verfahren, das orthogonale Aufwärtszeichnungen in  $\mathcal{O}(n \log \log n)$  Fläche (Höhe  $\mathcal{O}(n \log \log n / \log n)$ , Breite  $\mathcal{O}(\log n)$ ) mit höchstens  $\mathcal{O}(n)$  Kantenknicken erstellt [GGT93, GGT96]. Binärbäume können in strengen Aufwärtszeichnungen in linearer Fläche mit nur  $\mathcal{O}(n/\log n)$  Knicken gezeichnet werden [Kim95]. Darin wurde ebenfalls bewiesen, dass Ternärbäume auf dem 4-er Gitter in orthogonalen Aufwärtszeichnungen in  $\Theta(n \log n)$  Fläche gezeichnet werden können. Ein anderer ähnlicher Ansatz ist es, sogenannte *tip-over* Zeichnungen zu erstellen, siehe dazu [ELL93]. Es ist zu bemerken, dass strenge Aufwärtszeichnungen von Ternärbäumen auf dem 4-er Gitter nicht möglich sind, da ein Knoten bis zu drei Kinder haben kann, was nicht mehr streng monoton steigend bezüglich der  $y$ -Achse wäre.

### 3.1.3.3 Geradlinig

In diesem Abschnitt werden Zeichnungen behandelt, in denen die Kanten nicht auf dem Gitter verlaufen, sondern geradlinige Verbindungen zwischen beliebigen Knotenpaaren darstellen. Für strenge Aufwärtszeichnungen vollständiger Binärbäume gibt Trevisan eine obere Schranke für die Fläche in  $n + \mathcal{O}(\log n \sqrt{n})$  an [Tre96]. Die Idee ist, die Zeichnung in zwei Teile zu zerlegen. Zuerst werden in einer trivialen die Level 0 bis  $\lfloor (h+1)/2 \rfloor$  gezeichnet, um dann sukzessive die  $2^{\lfloor (h+1)/2 \rfloor}$  Unterbäume der Höhe  $h/2$  zu zeichnen. Diese Zeichnungen ähneln dem hierarchischen Ansatz, wobei hier nicht die Ästhetikkriterien AK1 - AK4 beachtet werden, siehe Abschnitt 3.1.1. Des Weiteren präsentiert er ein Verfahren für die obere Schranke für *Fibonacci-Bäume* in  $1.171n + \mathcal{O}(\log n \sqrt{n})$  Fläche. 1999 wurde von Chan ein Verfahren vorgestellt, das beliebige geordnete Bäume streng aufwärts, ordnungserhaltend und geradlinig in der Fläche  $\mathcal{O}(n^{1+\epsilon})$  mit  $\epsilon > 0$  zeichnet [Cha99]. Die beiden letzteren Ansätze sind auf Binärbäume, sowie auch auf  $d$ -näre Bäume anwendbar [CP98].

Ein *logarithmischer Baum* hat die Eigenschaft, dass die Höhe jedes genügend hohen Unterbaums logarithmisch hinsichtlich der Anzahl der Knoten ist [CP98]. Für Binär-, Ternär- und  $d$ -näre Bäume zeigen Crescenzi und Penna, dass die benötigte Fläche von ordnungserhaltenden strengen Aufwärtszeichnungen von geordneten logarithmischen Bäumen in  $\mathcal{O}(n)$  Fläche liegt.

### 3.1.3.4 Polyline

Zuletzt werden in diesem Abschnitt Zeichnungen von Bäumen betrachtet, deren Kanten Knicke auf ganzzahligen Koordinaten haben dürfen und die nicht auf dem Gitter verlaufen. Es gibt Linearzeit-Verfahren, die für ungeordnete Binärbäume Aufwärtszeichnungen in linearer Fläche mit der Breite  $\mathcal{O}(n^\alpha)$  und der Breite  $\mathcal{O}(n^{1-\alpha})$  ( $\alpha \in ]0, 1[$ ) erreichen [GGT93, GGT96]. Des Weiteren wird gezeigt, dass für aufwärts gezeichnete, ordnungserhaltende Zeichnungen von Binärbäumen die untere Schranke

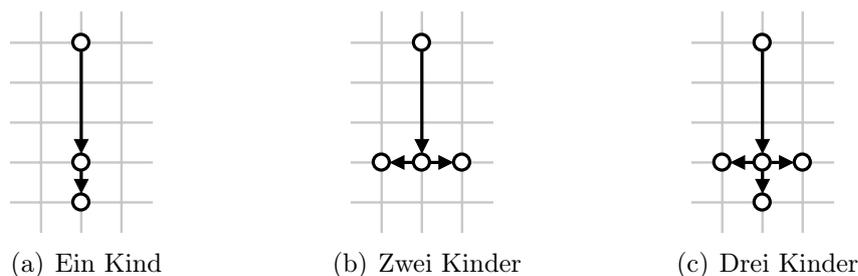


Abbildung 3.14. Pattern auf dem 4-er Gitter

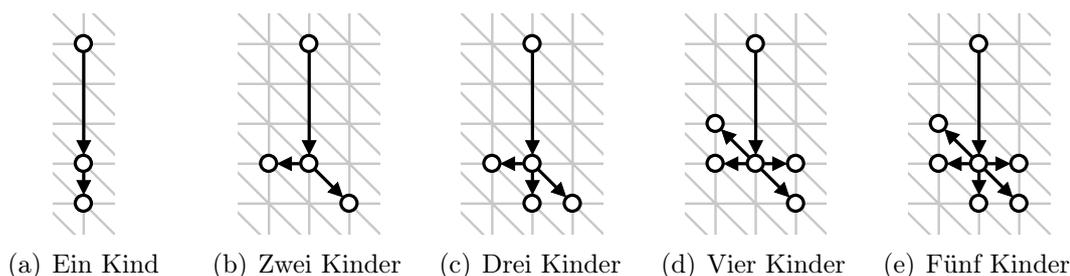


Abbildung 3.15. Pattern auf dem 6-er Gitter

$\Omega(n \log n)$  gilt.

Im nächsten Abschnitt werden neue Ergebnisse auf dem 4-er Gitter und dem 6-er und 8-er Gitter präsentiert. Dabei wird nur auf geradlinige Gitterzeichnungen und Gitterzeichnungen, deren Kanten auf den Kanten des Gittergraphens eingebettet sind, eingegangen. Zuerst wird auf vollständige  $d$ -näre Bäume eingegangen mit  $d < k$  auf einem  $k$ -Gitter mit  $k \in \{4, 6, 8\}$ . Danach werden geordnete und ungeordnete Bäume bezüglich verschiedener Ästhetikkriterien gezeichnet.

## 3.2 Neue Ansätze

In diesem Kapitel werden Zeichnungen von Bäumen auf dem 4-er, 6-er und 8-er Gitter diskutiert. Dazu wird nach Definitionen von Ästhetikkriterien zunächst auf vollständige Bäume eingegangen und es werden Schranken für die benötigten Flächen bestimmt. Danach werden Flächenschranken für ordnungserhaltende Zeichnungen und die Komplexität des Bestimmens der minimalen Fläche für verschiedene Ästhetikkriterien diskutiert. Zuletzt wird die Komplexität des Berechnens der minimalen Fläche und dem Zeichnen von Bäumen mit einheitlichen Kantenlängen für ungeordnete Bäume erörtert.

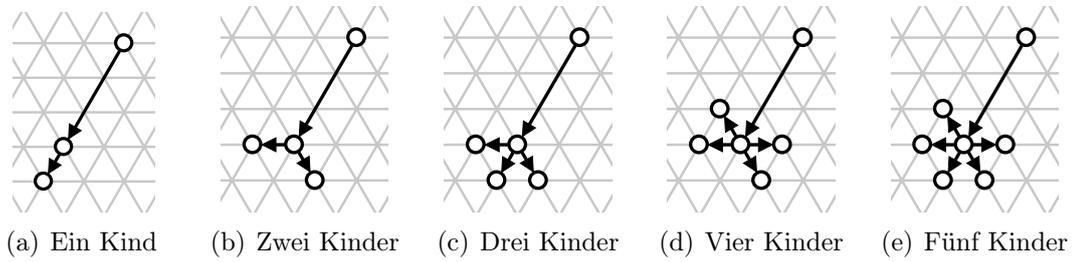
Mit der Erweiterung des 4-er Gitters ergeben sich für die Kantenrichtungen weitere Möglichkeiten. Betrachte dazu eine Baum  $T = (V, E)$  und den Knoten  $v \in V$

mit einer ausgehenden Kante  $(v, v_1)$  zum einzigen Kind  $v_1$ . Auf dem 4-er Gitter gibt es genau drei Möglichkeiten für die Richtung der geradlinigen Kante  $(v, v_1)$ . Auf dem 6-er (8-er) Gitter ergeben sich mehr Möglichkeiten, nämlich fünf (sieben) mögliche Richtungen.

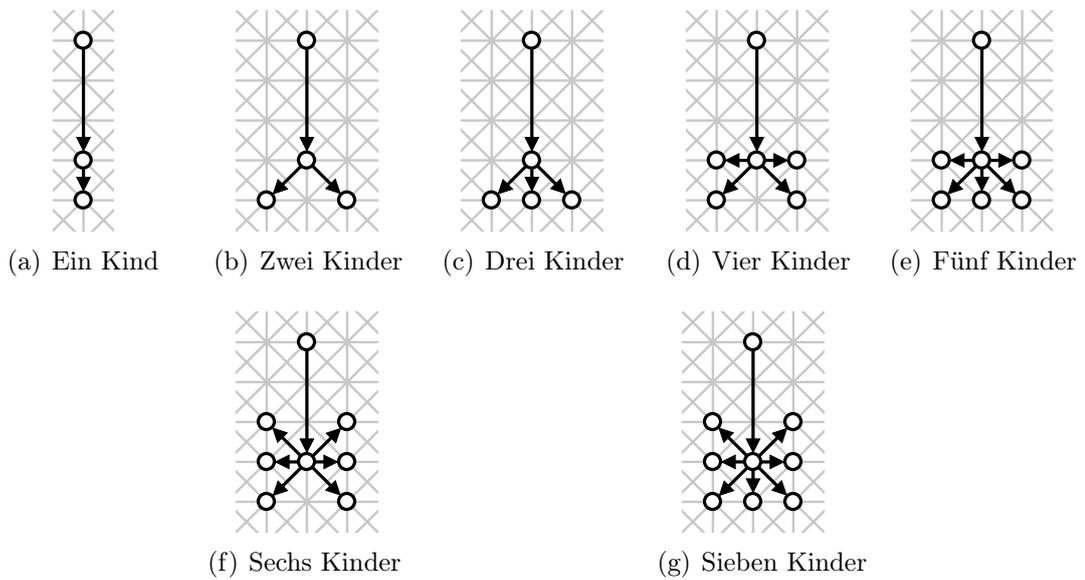
Betrachte einen Knoten  $v \in V$  mit einer eingehenden Kante  $(u, v)$  und einer ausgehenden Kante zum Knoten  $v_1$ . Würde die Kante  $(v, v_1)$  zur eingehenden Kante  $(u, v)$  einen Winkel von  $\pi/2$ ,  $\pi/3$  oder  $\pi/4$  einschließen, so könnte der Betrachter die Vater-Kind Beziehung nicht so leicht erkennen als wenn die Kante  $(v, v_1)$  die gleiche Richtung besitzt wie die eingehende Kante von  $v$ . Somit ist es aus ästhetischer Sicht schöner, wenn die Kante  $(v, v_1)$  die gleiche Richtung besitzt wie die eingehende Kante von  $v$ , siehe Abbildungen 3.14(a), 3.15(a), 3.16(a), 3.17(a). Aus diesem Gesichtspunkt kann man auch die ausgehenden Kanten eines Knotens  $v \in V$  mit zwei Kindern  $v_1$  und  $v_2$  betrachten. Die Richtung der eingehenden Kante von  $v$  soll in etwa beibehalten werden. Auch die Symmetrie relativ zur eingehenden Kante soll soweit wie möglich erhalten bleiben, so dass sich dadurch auf dem 4-er Gitter das zweite Pattern (Abbildung 3.14(b)) ergibt. Auf dem symmetrischen 6-er Gitter schließen demzufolge die ausgehenden Kanten und die eingehende Kante paarweise einen Winkel von  $2/3\pi$  ein, was dem Buchstaben  $Y$  ähnelt und deshalb als  $Y$ -Pattern bezeichnet wird, siehe Abbildungen 3.15(b) und 3.16(b). Analog dazu wird das  $Y$ -Pattern (Abbildung 3.17(b)) für das 8-er Gitter definiert. Hat ein Knoten  $v$  drei Kinder  $v_1, v_2$  und  $v_3$  so sind die Richtungen auf dem 4-er Gitter festgelegt, da die eingehende Kante eine Richtung blockiert, siehe Abbildung 3.14(c). Für das 6-er und 8-er Gitter wird wieder aus Gründen der Symmetrie und Richtung der eingehenden Kante von  $v$  eine ausgehende Kante geradeaus gezeichnet und die anderen beiden Kanten belegen die Richtungen daneben. Dieses Pattern für Knoten mit drei Kindern ähnelt dem Zeichen  $\Psi$  und wird demzufolge  $\Psi$ -Pattern genannt, siehe Abbildungen 3.15(c), 3.16(c) und 3.17(c). Hat ein Knoten vier Kinder, so kann in einer Zeichnung auf dem 6-er Gitter das sogenannte  $X$ -Pattern verwendet werden. Dabei werden alle Richtungen für die ausgehenden Kanten zu den Kindern verwendet, außer die Richtung der eingehenden Kante und die Richtung entgegengesetzt dazu, siehe Abbildungen 3.15(d) und 3.16(d). Die restlichen Pattern für das 6-er und das 8-er Gitter werden analog definiert, siehe Abbildung 3.15 für das 6-er Gitter, Abbildung 3.17 für das 8-er Gitter und siehe Abbildung 3.16 für das symmetrische 6-er Gitter.

**Definition 3.7 (Pattern-Zeichnung).** *In einer Pattern-Zeichnung eines  $d$ -nären Baumes  $T = (V, E)$  mit  $d < k$  auf dem  $k$ -Gitter haben die ausgehenden Kanten jedes Knotens  $v \in V$  die soeben beschriebenen Winkel relativ zur Richtung der eingehenden Kante von  $v$ .*

In Abschnitt 3.1.1 wurden für hierarchische Zeichnungen von Bäumen die Ästhetikkriterien  $AK1$  bis  $AK4$  vorgestellt. Darunter waren auch Kriterien, welche erforderten, dass Knoten desselben Levels auf einer Geraden platziert werden sollen und, dass der Vaterknoten zentriert über den entsprechenden Söhnen gezeichnet



**Abbildung 3.16.** Pattern auf dem symmetrischen 6-er Gitter



**Abbildung 3.17.** Pattern auf dem 8-er Gitter

werden soll. Versucht man diese beiden Kriterien auf das geradlinige Zeichnen von Bäumen auf einem Gitter zu übertragen, ergeben sich folgende Definitionen:

**Definition 3.8 (Global uniforme Zeichnung).** *Eine Zeichnung eines  $d$ -nären Baumes auf dem  $k$ -Gitter mit  $d < k$  ist global uniform, wenn für die Knoten desselben Levels gilt, dass die ausgehenden Kanten die gleiche Länge besitzen.*

Es ist zu beachten, dass in global uniformen Zeichnungen nicht nur Geschwisterknoten denselben Abstand zum Vaterknoten haben, sondern alle Knoten desselben Levels denselben Abstand zu den entsprechenden Vaterknoten. Diese Eigenschaft gibt den Knoten desselben Levels die gleiche Priorität. Es verhindert, dass ein Knoten aufgrund der gleichen Kantenlängen gegenüber seinen Geschwisterknoten oder anderen Levelknoten stärker oder schwächer wahrgenommen wird.

**Definition 3.9 (Lokal uniforme Zeichnung).** *Eine Zeichnung eines  $d$ -nären Baumes  $T = (V, E)$  auf dem  $k$ -Gitter mit  $d < k$  ist lokal uniform, wenn für jeden Knoten  $v \in V$  gilt, dass die ausgehenden Kanten von  $v$  die gleiche Länge besitzen.*

Die Definition der lokalen Uniformität schwächt die globale Uniformität so ab, dass nur noch die eingehenden Kanten der Geschwisterknoten gleiche Länge besitzen müssen, jedoch nicht mehr alle eingehenden Kanten eines bestimmten Levels.

Bezieht man sich jetzt auf die Ästhetikkriterien aus Abschnitt 3.1.1, speziell auf Vater zentriert über den Söhnen und Knoten desselben Levels auf derselben Gerade, dann sind die Ästhetikkriterien Pattern und lokale (globale) Uniformität sinnvoll erweitert. Aufgrund der globalen Uniformität der ausgehenden Kanten eines Knotens bekommt jedes Kind die gleiche Priorität wie im hierarchischen Fall.

Bloesch hat schon für rechteckige Knoten gefordert, dass Geschwisterknoten denselben Abstand zum Vaterknoten besitzen sollen [Blo93], siehe Abbildung 3.9. Diese Forderung ähnelt der Forderung nach lokaler Uniformität. Des Weiteren hat Walker für  $d$ -näre Bäume nur die Reihenfolge der Kinder aus dem Ansatz von Reingold und Tilford übernommen [Wal90, RT81], so dass im hierarchischen Fall die Kante eines Kindes ohne Geschwisterknoten vertikal gezeichnet wird. Dies wird auch hier von entsprechenden Pattern übernommen. Die Wahl dieser Ästhetikkriterien kann aber zusätzlich auch durch die bessere Lesbarkeit von Zeichnungen begründet werden. Betrachte dazu die Beispiele in Abbildung 3.18. In Abbildung 3.18(a) kann die Struktur des Baumes nicht leicht erkannt werden, weil die Kantenrichtungen zufällig gewählt wurden und die ausgehenden Kanten eines Knotens nicht unbedingt die gleiche Länge besitzen. Es ist schwierig Knoten desselben Levels und die Anzahl der Kinder eines Knotens zu bestimmen. Nimmt man Abbildung 3.18(b), so haben zwar die ausgehenden Kanten jedes Knotens dieselbe Länge, aber trotzdem ist diese Zeichnung nicht so übersichtlich wie Abbildung 3.18(d), die eine lokal uniforme Pattern-Zeichnung darstellt. Eine Pattern-Zeichnung ist in Abbildung 3.18(c) gezeigt. Die ausgehenden Kanten haben die durch die Pattern vorgegebene Richtung, was die Zeichnung im Vergleich zu Abbildung 3.18(a) etwas übersichtlicher macht. Aber Geschwisterknoten haben darin nicht gleiche Entfernung zu ihrem Vater. In

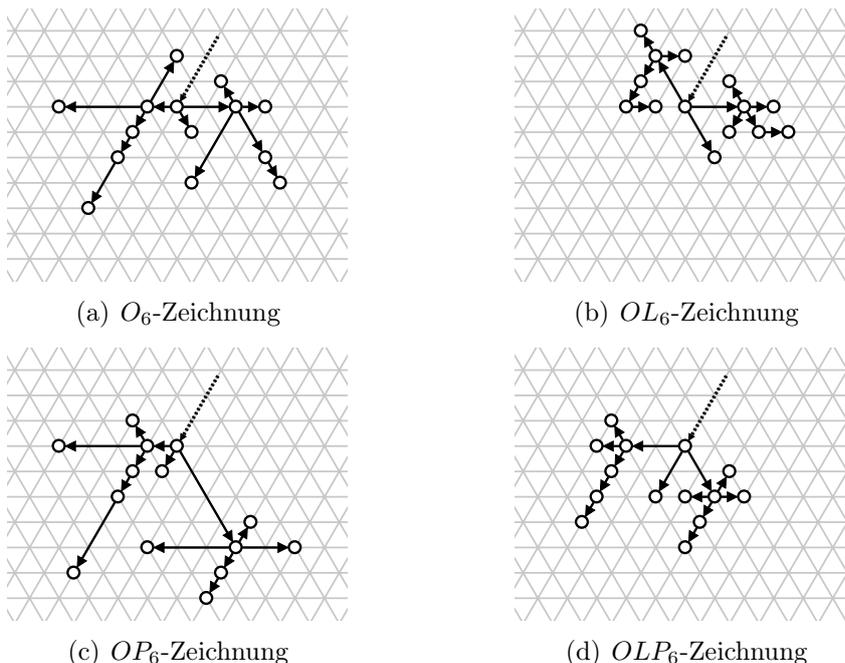


Abbildung 3.18. Beispiele auf dem symmetrischen 6-er Gitter

Pattern-Zeichnungen haben die ausgehenden Kanten von Knoten mit gleichem Ausgangsgrad immer die gleichen Richtungen relativ zur Richtung der eingehenden Kante. Somit kann leichter erkannt werden, wieviele Kinder ein Knoten besitzt.

### 3.2.1 Vollständige Bäume

In diesem Abschnitt werden vollständige  $d$ -näre Bäume mit  $d < k$  auf  $k$ -Gittern mit  $k \in \{4, 6, 8\}$  behandelt. Im Vergleich zum 4-er Gitter mit bis zu vier inzidenten Kanten pro Knoten erlaubt das 6-er (8-er) Gitter bis zu sechs (acht) inzidente Kanten. Es werden Schranken für das Zeichnen von vollständigen  $d$ -nären Bäumen und Ternärbäumen auf dem  $k$ -Gitter diskutiert. Schließlich werden vollständige Ternärbäume und vollständige 7-näre Bäume auf dem 8-er Gitter behandelt.

In Tabelle 3.2 sind die Schranken für  $d < k$  und  $k \in \{4, 6, 8\}$  aufgelistet. Für Binärbäume wurde die obere und untere Schranke auch für  $hv$ -Zeichnungen bewiesen [CDBP92], somit gelten diese Schranken auch für das 6-er und das 8-er Gitter, da das 4-er Gitter eine Teilmenge dieser beiden Gitter darstellt. Dass Ternärbäume in der Fläche  $\mathcal{O}(n^{1.262})$  gezeichnet werden können, wurde von Frati gezeigt [Fra07]. Die Fälle für  $d \geq k$  sind nicht möglich, da diese Bäume, speziell die Kanten, nicht planar in ein Gitter eingebettet werden können. Die restlichen Ergebnisse werden im Folgenden aufgeführt und gezeigt.

**Satz 3.1.** *Es existiert ein Linearzeitalgorithmus zur Konstruktion einer global uniformen Zeichnung eines vollständigen 5-nären Baumes mit  $n$  Knoten auf dem 6-*

	$k = 4$	$k = 6$	$k = 8$	
$d = 2$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	[CDBP92]
$d = 3$	$\mathcal{O}(n^{1.262})$	$\Theta(n^{1.262})$	$\mathcal{O}(n^{1.048})$	[Fra07], Satz 3.3, 3.7
$d = 4$	—	$\mathcal{O}(n^{1.585})$	$\mathcal{O}(n^{1.585})$	Kor. 3.1, Kor. 3.5
$d = 5$	—	$\mathcal{O}(n^{1.366})$	$\mathcal{O}(n^{1.240})$	Satz 3.1, Kor. 3.4
$d = 6$	—	—	$\mathcal{O}(n^{1.227})$	Kor. 3.3
$d = 7$	—	—	$\Theta(n^{1.130})$	Satz 3.8

**Tabelle 3.2.** Flächenschranken für vollständige  $d$ -näre Bäume auf dem  $k$ -Gitter

*Gitter in  $\mathcal{O}(n^{1.366})$  Fläche und mit Seitenverhältnis 1.*

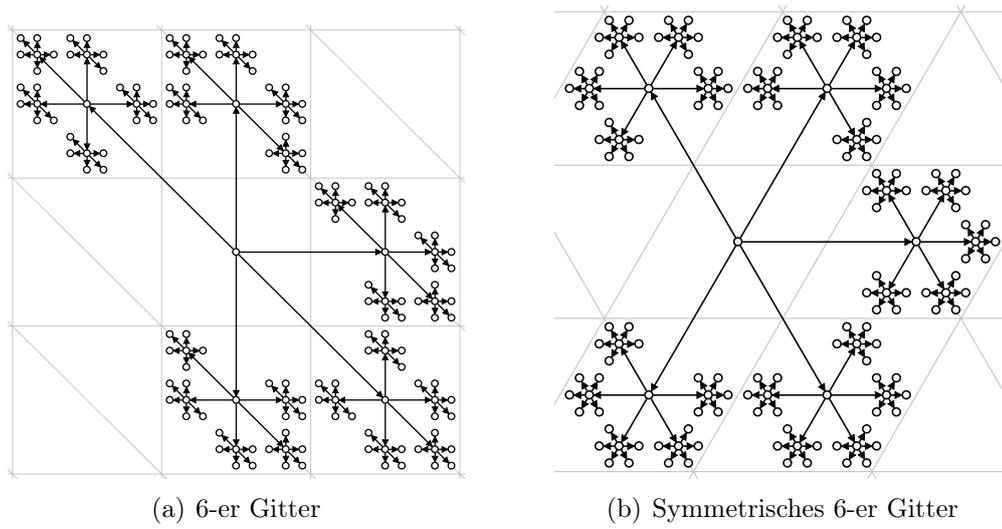
*Beweis.* Die Zeichnungen werden rekursiv erstellt. Durch die Verdreifachung in jede Richtung kann man einen neuen Baum der Höhe  $h + 1$  planar zeichnen, siehe Abbildung 3.19(a). Somit ist die Fläche in  $\mathcal{O}(9^h)$ , wobei  $h$  die Höhe des Baumes ist. Alle Kanten derselben Tiefe haben dieselbe Länge und weil  $h = \log_3 n$  gilt, ist die Fläche in  $\mathcal{O}(9^{\log_3 n}) = \mathcal{O}(n^{\log_3 9})$ , was in  $\mathcal{O}(n^{1.366})$  ist.  $\square$

Die Beispielzeichnung desselben Baumes aus Abbildung 3.19(a) ist in Abbildung 3.19(b) auf dem symmetrischen 6-er Gitter gezeigt, die aufgrund der besseren Symmetrieeigenschaft gleichmäßiger wirkt als die Zeichnung in Abbildung 3.19(a). In der gleichen Art und Weise können vollständige 4-näre Bäume in der Fläche  $\mathcal{O}(n^{\log_4 9})$  auf dem 6-Gitter gezeichnet werden, vgl. Abbildung 3.20. Eine *X-Pattern-Zeichnung* ist eine Zeichnung eines Baumes, in dem jeder Knoten entweder vier Kinder hat oder ein Blatt ist und immer das X-Pattern in der Zeichnung angewendet wird.

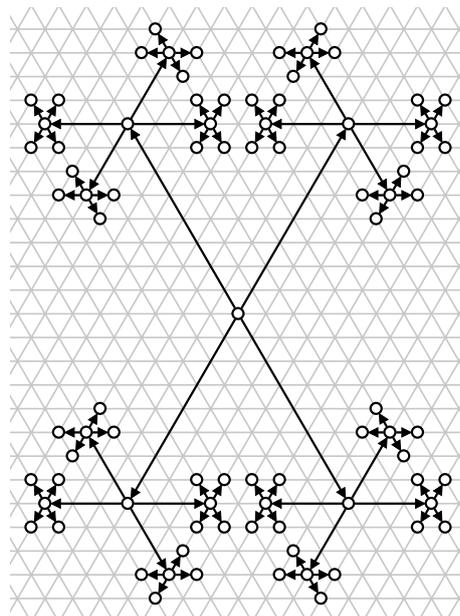
**Korollar 3.1.** *Es existiert ein Linearzeitalgorithmus zum Zeichnen einer global uniformen X-Pattern-Zeichnung eines vollständigen 4-nären Baumes auf dem 6-Gitter in  $\mathcal{O}(n^{1.585})$  Fläche und mit Seitenverhältnis 1.*

*Beweis.* Die Zeichnungen werden analog zum Beweis zu Satz 3.1 rekursiv erstellt. Die Fläche ist wieder in  $\mathcal{O}(9^h)$ , wobei  $h$  die Höhe des Baumes ist. Alle Kanten derselben Tiefe haben dieselbe Länge und weil  $h = \log_4 n$ , ist die Fläche in  $\mathcal{O}(n^{\log_4 9})$ , was in  $\mathcal{O}(n^{1.585})$  ist.  $\square$

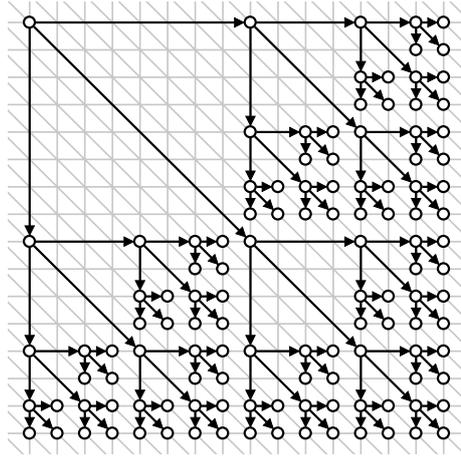
Als nächstes werden strenge Aufwärtszeichnungen von vollständigen Ternärbäumen auf dem 6-er Gitter betrachtet. Dieser Ansatz ähnelt  $hv$ -Zeichnungen, die auf dem 4-er Gitter eingebettet werden, siehe Abschnitt 3.1.3.1. Die ausgehenden inzidenten Kanten eines inneren Knotens verlaufen entweder horizontal nach rechts, vertikal nach unten oder diagonal nach rechts unten. Somit sind Zeichnungen mit



**Abbildung 3.19.** Lokal uniforme Pattern-Zeichnung eines vollständigen 5-nären Baumes



**Abbildung 3.20.** Lokal uniforme Pattern-Zeichnung eines vollständigen 4-nären Baumes auf symmetrischem 6-er Gitter



**Abbildung 3.21.**  $hvd_6$ -Zeichnung eines vollständigen Ternärbaumes

dieser Eigenschaft strenge Aufwärtszeichnungen. Diese Zeichnungen ähneln  $hvd$ -Zeichnungen [CP98], wobei die Steigung der Kanten auf der Diagonalen auf dem 6-er Gitter fest durch die Diagonale vorgegeben ist. Aufgrund der Ähnlichkeit der Zeichnungen zu den  $hvd$ -Zeichnungen werden die Zeichnungen auf dem 6-er Gitter  $hvd_6$ -Zeichnungen genannt, siehe Abbildung 3.21. Für  $hvd_6$ -Zeichnungen wird im Folgenden eine feste obere und untere Schranke für die Höhe und Breite und somit für die Fläche gezeigt.

**Lemma 3.1.** *Jede  $hvd_6$ -Zeichnung eines vollständigen Ternärbaumes mit  $n$  Knoten hat mindestens eine Breite (und Höhe) in  $\Omega(n^{0.631})$ .*

*Beweis.* Betrachte eine  $hvd_6$ -Zeichnung  $\Gamma(T)$  eines vollständigen Ternärbaumes  $T$  mit der Höhe  $h$  auf dem 6-Gitter. Es wird gefordert, dass die Gitterpunkte  $(0, 2^h - 1)$  und  $(2^h - 1, 0)$  durch die Zeichnung belegt sind. Somit befinden sich auf diesen Gitterpunkten entweder ein Knoten oder darüber verläuft eine Kante. Im Beweis wird über  $h$  induziert. Für  $h = 1$  ist die Aussage trivialerweise korrekt.

Angenommen, es existiert ein minimales  $h$ , so dass ein vollständiger Unterbaum der Höhe  $h$  diese beiden extremen Gitterpunkte nicht belegt. Sei  $r$  die Wurzel eines Baumes der Höhe  $h$  mit den Koordinaten  $(0, 0)$  und seien  $T_1, T_2$  und  $T_3$  Unterbäume der Kinder von  $r$ . Jede  $hvd_6$ -Zeichnung eines Baumes der Höhe  $h - 1$  belegt nach Induktionsannahme die Punkte  $(2^{h-1} - 1, 0)$  und  $(0, 2^{h-1} - 1)$  relativ zu ihrer Wurzel. Es existiert eine vertikale Linie von  $r$  zu  $T_3$ , eine diagonale Linie von  $r$  zu  $T_1$  und eine horizontale Linie von  $r$  zu  $T_2$ . Wenn  $T_3$  den Gitterpunkt  $(0, 2^h - 1)$  nicht belegt, dann muss er einen Gitterpunkt  $(p, p)$  mit  $p \leq 2^{h-1} - 1$  auf der Diagonalen belegen. Damit wäre aber für  $T_1$  kein Platz mehr. Mit dem symmetrischen Argument belegt  $T_2$  den Gitterpunkt  $(2^h - 1, 0)$ . Da sich die Höhe (Breite) von  $h$  nach  $h + 1$  verdoppelt und die Anzahl der Knoten sich mindestens verdreifacht ergibt sich für die Höhe (Breite)  $\Omega(2^h) = \Omega(2^{\log_3 n}) = \Omega(n^{\log_3 2})$ . Daraus resultiert die untere Schranke für die Höhe und Breite in  $\Omega(n^{0.631})$ .  $\square$

**Satz 3.2.** *Es existiert ein Linearzeitalgorithmus zum Zeichnen von global uniformen  $hvd_6$ -Zeichnungen von vollständigen Ternärbäumen mit  $n$  Knoten auf dem 6-Gitter in der Fläche  $\mathcal{O}(n^{1.262})$  und Seitenverhältnis 1.*

*Beweis.* Die Zeichnung wird rekursiv konstruiert, so dass die Wurzel in der linken oberen Ecke ist. Die drei Unterbäume der Höhe  $h$  sind jeweils innerhalb eines Quadrats mit Höhe und Breite  $S(h)$  gezeichnet (siehe Abbildung 3.21). Die drei Unterbäume werden um  $1+S(h)$  Einheiten ausgehend von der Position der Wurzel entlang der  $X$ -,  $Y$ -Achse und der Diagonalen verschoben. Dadurch entsteht eine Zeichnung mit Höhe und Breite  $S(h+1) = 2S(h) + 1$ . Hieraus ergibt sich für  $S(h) = 2^h - 1$  und eine global uniforme Zeichnung. Da  $h = \log_3 n$ , erhält man für einen vollständigen Ternärbaum mit  $n$  Knoten eine Zeichnung mit Höhe und Breite in  $\mathcal{O}(n^{0.631})$  und somit in der Fläche  $\mathcal{O}(n^{1.262})$ .  $\square$

Somit ergibt sich aus Lemma 3.1 und Satz 3.2 der folgende Satz hinsichtlich der oberen und unteren Schranke für  $hvd_6$ -Zeichnungen auf dem 6-er Gitter.

**Satz 3.3.** *Die obere und untere Schranke für die Fläche einer  $hvd_6$ -Zeichnung eines vollständigen Ternärbaumes mit  $n$  Knoten im 6-er Gitter ist in  $\Theta(n^{1.262})$ .*

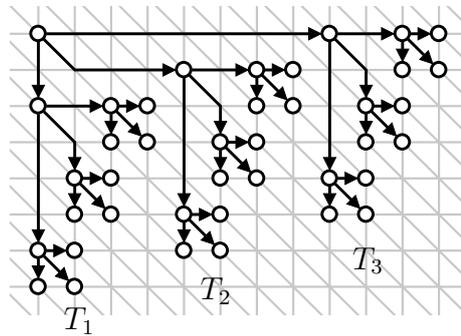
Sollen die Kanten weiterhin nach rechts, unten und diagonal nach rechts unten verlaufen mit dem Zusatz, dass Knicke erlaubt werden, dann ergibt sich nachfolgender Satz.

**Satz 3.4.** *Es existiert ein Linearzeitalgorithmus zum Zeichnen eines vollständigen Ternärbaumes mit höchstens einem Knick pro Kante auf dem 6-er Gitter in der Fläche  $\Theta(n)$ . Die Zeichnung ist streng aufwärts planar, hat ein Seitenverhältnis in  $\mathcal{O}(1)$  und insgesamt weniger als  $\frac{n}{9}$  Knicke.*

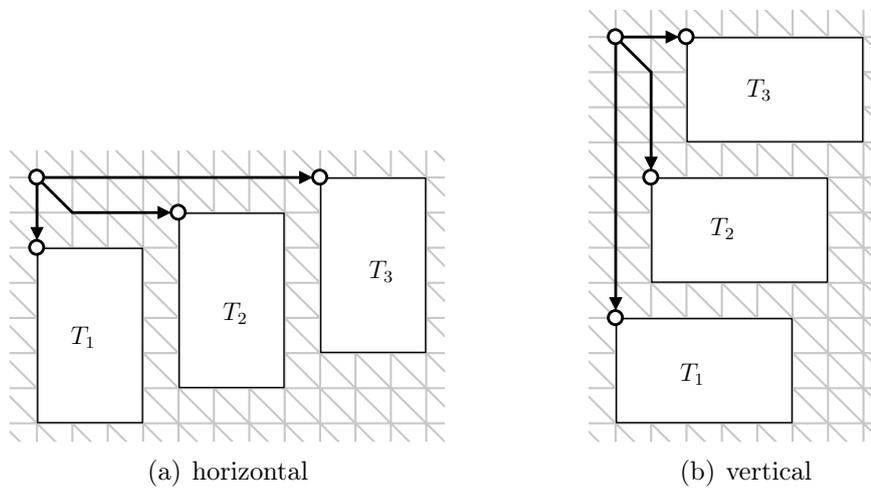
*Beweis.* Eine Zeichnung eines vollständigen Ternärbaumes  $T(E)$  der Höhe  $h$  wird wie folgt konstruiert. Es wird rekursiv gezeichnet von  $h \rightarrow h+1$  mit Ternärbäumen der Höhe  $h$ . Diese werden im Rekursionsschritt verwendet, um einen Ternärbaum der Höhe  $h+1$  zu zeichnen. Für  $h=0$  (ein Knoten) und  $h=1$  (ein innerer Knoten und drei Blätter) ist die Zeichnung offensichtlich, siehe Abbildung 3.22.

Der Konstruktionsschritt sieht für  $h \bmod 2 = 1$  einen horizontalen Kompositionsschritt vor (Abbildung 3.23(a)). Drei vollständige Ternärbäume  $T_1, T_2$  und  $T_3$  mit entsprechenden Wurzelknoten  $r_1, r_2$  und  $r_3$  der Höhe  $h$  ( $h$  ungerade) sind mit Höhe  $H$  und Breite  $W$  gezeichnet und werden nebeneinander platziert. Der Wurzelknoten  $r_1$  des Ternärbaumes  $T_1$  wird auf Koordinate  $(0, 2)$  platziert,  $r_2$  auf  $(W+1, 1)$  und  $r_3$  auf  $(2W+1, 0)$ . Für  $h \bmod 2 = 0$  ( $h$  gerade) wird ein vertikaler Kompositionsschritt (Abbildung 3.23(b)) vorgenommen. Dabei werden  $T_1, T_2$  und  $T_3$  untereinander platziert mit  $r_1$  auf  $(0, 2H+1)$ ,  $r_2$  auf  $(1, H+1)$  und  $r_3$  auf  $(2, 0)$ .

Im Konstruktionsschritt kann man erkennen, dass sich die Zeichnung im horizontalen Kompositionsschritt ( $h \bmod 2 = 1$ ) in der Breite verdreifacht und um zwei verbreitert,  $W(h+1) = 3 \cdot W(h)$  und  $H(h+1) = H(h) + 2$ . Für gerades  $h$  ist es genau entgegengesetzt:  $W(h+1) = W(h) + 2$  und  $H(h+1) = 3 \cdot H(h)$ .



**Abbildung 3.22.** Beispielzeichnung eines vollständigen Ternärbaumes mit weniger als  $n/9$  Knicken auf dem 6-er Gitter



**Abbildung 3.23.** Konstruktionsschritte für den Beweis von Satz 3.4

Wenn man die Rekursion auflöst, so ergeben sich für die Höhe  $H$  und die Breite  $W$  der konstruierten Zeichnung dann folgende Funktionen  $H(h)$  und  $W(h)$  mit  $h \in \mathbb{N}^+$  und  $h > 1$ . Dabei wird die Höhe und die Breite in der Anzahl der Gitterpunkte betrachtet.

$$H(h) = 3 \cdot 3^{\lfloor \frac{h}{2} \rfloor} - 2 - (-1)^h \quad (3.1)$$

$$W(h) = 5 \cdot 3^{\lfloor \frac{h}{2} \rfloor} - 2 + (-1)^h \quad (3.2)$$

Die Korrektheit der Funktionen kann mittels Induktion gezeigt werden. Da für einen vollständigen Ternärbaum der Höhe  $h$  für die Anzahl der Knoten  $n$  gilt:  $n = \frac{3^{h+1}-1}{2}$ , kann hinsichtlich der  $\mathcal{O}$ -Notation für  $h$  den Wert  $h = \log_3 n$  in die Funktionen  $H$  und  $W$  eingesetzt werden:

$$H(n) \in \mathcal{O}(3 \cdot 3^{\lfloor \frac{\log_3 n}{2} \rfloor} - 2 - (-1)^{\log_3 n}) \quad (3.3)$$

$$W(n) \in \mathcal{O}(5 \cdot 3^{\lfloor \frac{\log_3 n}{2} \rfloor} - 2 + (-1)^{\log_3 n}) \quad (3.4)$$

$$H(n) \in \mathcal{O}(\sqrt{n}) \quad (3.5)$$

$$W(n) \in \mathcal{O}(\sqrt{n}) \quad (3.6)$$

Die untere Schranke ist trivial und somit gilt für die Fläche  $A = H \cdot W \in \Theta(n)$ , mit Seitenverhältnis in  $\mathcal{O}(1)$ .  $\square$

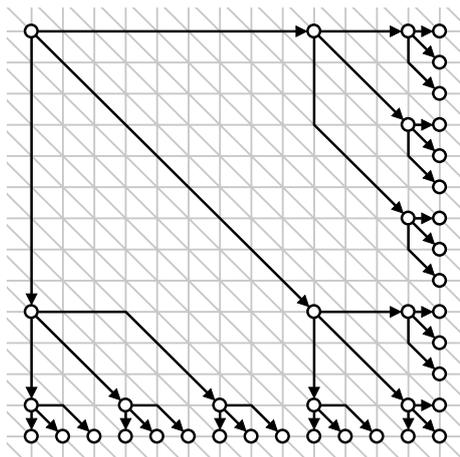
Wie schon in Abschnitt 3.1.3.2 erwähnt, ist es in der Computerchipentwicklung oft notwendig Anschlusspunkte am Rand der Chips zu platzieren [Val81, PRS81, BK80, BL83, Lei84]. Dieser Ansatz wird nachfolgend für Zeichnungen auf dem 6-er Gitter diskutiert, indem die Blätter eines Ternärbaumes am Rand der Zeichnung platziert werden.

Eine Zeichnung eines Baumes, in der alle Blätter auf dem Rand des kleinsten umschließenden Rechtecks platziert sind, wird *Leaves-Out Zeichnung* genannt.

**Satz 3.5.** *Eine streng aufwärts gerichtete Leaves-Out Zeichnung eines vollständigen Ternärbaumes auf dem 6-er Gitter kann in  $\mathcal{O}(n^2)$  Fläche mit weniger als  $n/3$  Knicken und einem Seitenverhältnis von 1 erstellt werden.*

*Beweis.* Aufgrund der Eigenschaft, dass die Blätter am Rand platziert werden und die Kanten nur nach rechts, nach unten und nach rechts unten verlaufen, sind die Blätter rechts vertikal und unten horizontal aufgereiht. Die Anzahl der |Blätter| in einem vollständigen Ternärbaum ist

$$|\text{Blätter}| = \left(2 \cdot \frac{n-1}{3}\right) + 1.$$



**Abbildung 3.24.** Vollständiger Ternärbaum mit Blättern außen auf dem 6-er Gitter

Somit ergibt sich für die Fläche

$$A = \left( \frac{|\text{Blätter}| - 1}{2} \right)^2 = \left( \frac{(2 \cdot \frac{n-1}{3} + 1) - 1}{2} \right)^2 = \left( \frac{n-1}{3} \right)^2 \in \mathcal{O}(n^2).$$

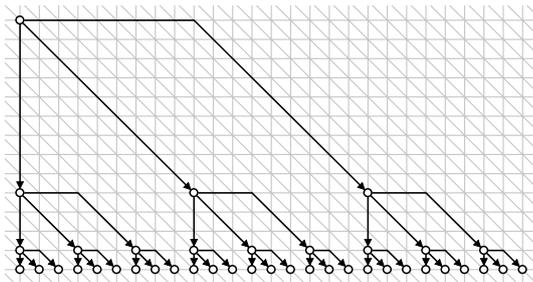
□

Eine weitere Spezialisierung von Leaves-Down Zeichnungen sind Zeichnungen, in der die Blätter alle am unteren Rand der Zeichnung platziert sind. Diese werden *Leaves-Down Zeichnungen* genannt.

**Satz 3.6.** *Eine streng aufwärts verlaufende Leaves-Down Zeichnung eines vollständigen Ternärbaumes auf dem 6-er Gitter kann in  $\mathcal{O}(n^2)$  Fläche mit weniger als  $n/3$  Knicken erstellt werden.*

*Beweis.* Sei  $T = (V, E)$  ein vollständiger Ternärbaum mit Höhe  $h$ . In der Zeichnung eines Baumes der Höhe  $h = 1$  mit vier Knoten wird die Wurzel  $r$  links oben auf die Koordinate  $(0, 0)$  platziert. Die Kinder  $u, v$  und  $w$  werden auf die Koordinaten  $(0, 1)$ ,  $(1, 1)$  und  $(2, 1)$  gesetzt. Die Kanten zu  $u$  und  $v$  werden geradlinig gezeichnet und für die Kante zu  $(r, w)$  wird ein Knick auf Koordinate  $(0, 1)$  eingefügt. Es ist zu beachten, dass die Wurzel  $r$  oben links platziert ist.

O. B. d. A. werden die Blätter am unteren Rand positioniert. Somit werden im Induktionsschritt  $h \rightarrow h + 1$  die drei gleich gezeichneten Unterbäume  $T_1, T_2$  und  $T_3$  mit der Höhe  $h$  nebeneinander platziert. Die Wurzel des Baumes  $T$  mit der Höhe  $h+1$  wird auf die Koordinate  $(0, 0)$  gesetzt. Die Koordinaten der Wurzeln  $r_1, r_2, r_3$  von  $T_1, T_2, T_3$  werden auf die Positionen  $(0, \text{width}(T_h) + 1)$ ,  $(\text{width}(T_h) + 1, \text{width}(T_h) + 1)$  und  $(2 \cdot \text{width}(T_h) + 2, \text{width}(T_h) + 1)$  gesetzt. Offensichtlich ergibt sich eine Breite in  $\mathcal{O}(n)$  und aufgrund der Induktion ebenfalls eine Höhe in  $\mathcal{O}(n)$ , womit sich eine Fläche in  $\mathcal{O}(n^2)$  ergibt. □



**Abbildung 3.25.** Leaves-Down Zeichnung eines vollständigen Ternärbaumes auf dem 6-er Gitter

Jeder vollständige Ternärbaum kann auf dem 4-er Gitter in der Fläche  $\mathcal{O}(n^{1.262})$  [Fra07] gezeichnet werden. Für strenge Aufwärtszeichnungen auf dem 6-er Gitter gibt es Flächenschranken in  $\Theta(n^{1.262})$ , siehe Satz 3.3. Im Folgenden wird ein Verfahren für eine obere Schranke für Ternärbäume auf dem 8-er Gitter angegeben, das alle acht möglichen Kantenrichtungen ausschöpft.

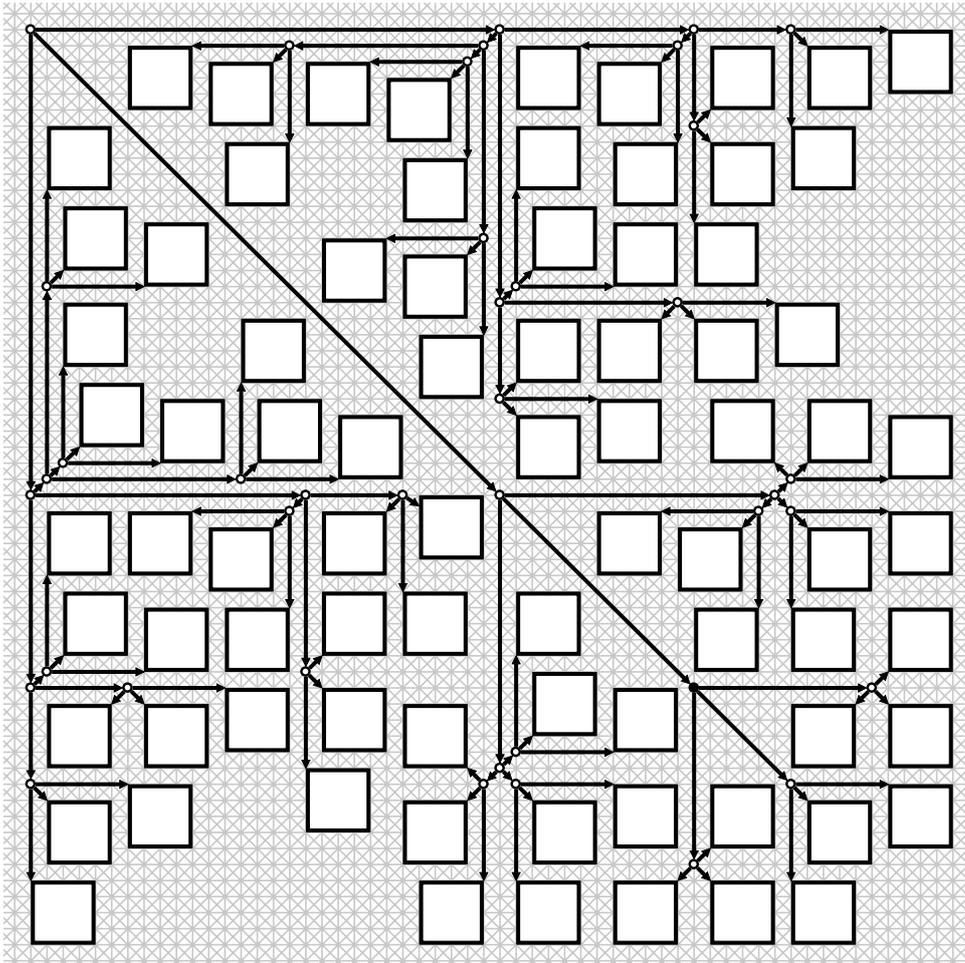
**Satz 3.7.** *Es gibt einen Linearzeitalgorithmus zum Zeichnen von vollständigen Ternärbäumen mit  $n$  Knoten auf dem 8-er Gitter in der Fläche  $\mathcal{O}(n^{\frac{\log_3 100}{4}}) \approx \mathcal{O}(n^{1.048})$  und mit Seitenverhältnis 1.*

*Beweis.* Dazu wird ein Baum  $T(h)$  mit der Höhe  $h$  rekursiv konstruiert. Abbildung 3.26 zeigt einen Konstruktionsschritt von  $i \rightarrow i + 4$  mit  $i \leq h$ . Initial wird für  $i = h \bmod 4 \in \{0, \dots, 3\}$  der Baum  $T(i)$  in ein Quadrat mit konstanter Fläche mit der Wurzel an einer der vier Ecken gezeichnet. Im Konstruktionsschritt von  $i \rightarrow i + 4$  gibt es 81 vollständige Bäume mit der Höhe  $i$ , die innerhalb eines Quadrates mit Seitenlänge  $S(i)$  gezeichnet sind. Sei  $c = 8$  die Anzahl, der im Konstruktionsschritt zusätzlich eingefügten Spalten bzw. Zeilen, die zum Verbinden der neuen inneren Knoten benötigt wird. Deshalb ist die Seitenlänge  $S(i + 4) = 10 \cdot S(i) + c \leq 10^{\lceil i/4 \rceil} + (c \cdot \sum_{i=0}^{\lceil i/4 \rceil} 10^i) < 10^{\lceil i/4 \rceil} + c \cdot 10^{\lceil i/4 \rceil + 1}$ .  $S(h) \in \mathcal{O}(10^{h/4})$  und die Fläche von  $T(h)$  ist in  $\mathcal{O}(100^{h/4})$ . Die Höhe eines vollständigen Ternärbaumes ist  $h = \log_3 n$ . Folglich ist die benötigte Fläche in  $\mathcal{O}(100^{(\log_3 n)/4}) = \mathcal{O}(n^{(\log_3 100)/4}) \subset \mathcal{O}(n^{1.048})$ .  $\square$

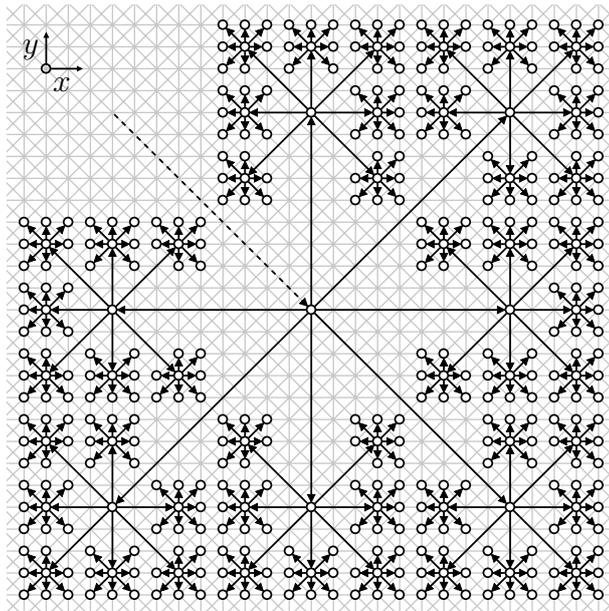
Nach der Betrachtung von Ternärbäumen auf dem 6-er und 8-er Gitter wird im Folgenden auf Zeichnungen von 7-nären Bäumen auf dem 8-er Gitter eingegangen und dazu Schranken für die benötigte Fläche angegeben.

**Satz 3.8.** *Die obere und die untere Schranke für die Fläche von Zeichnungen von vollständigen 7-nären Bäumen mit  $n$  Knoten auf dem 8-er Gitter ist in  $\Theta(n^{\log_7 9})$ .*

*Beweis.* Die Zeichnung wird rekursiv konstruiert. Der Initialfall ist ein Baum mit Höhe 0, also nur ein Knoten. Im Konstruktionsschritt  $h \rightarrow h + 1$  wächst die Seitenlänge der planaren Zeichnung um das dreifache, siehe Abbildung 3.27. Somit ist die



**Abbildung 3.26.** Zeichenschema für Satz 3.7 auf dem 8-er Gitter (Gitterkanten weggelassen)



**Abbildung 3.27.** Zeichnung eines vollständigen 7-nären Baumes auf dem 8-er Gitter (Gitterkanten weggelassen)

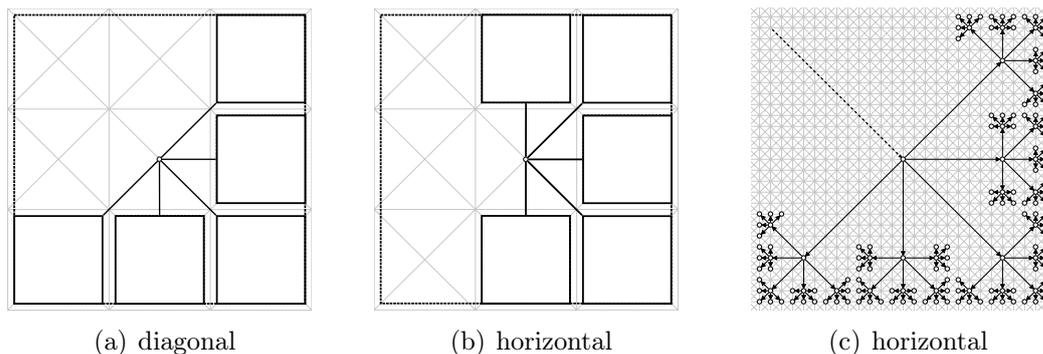
Fläche in  $\mathcal{O}(9^h)$  und, weil  $h = \log_7 n$  für balancierte Bäume gilt, ist die Fläche in  $\mathcal{O}(9^{\log_7 n}) = \mathcal{O}(9^{\log_7 n}) \approx \mathcal{O}(n^{1.130})$ .

Sei  $\Gamma(T(h))$  eine  $U_k$ -Zeichnung eines vollständigen 7-nären Baumes der Höhe  $h$  mit der Wurzel  $r$  auf dem 8-er Gitter. O. B. d. A. wird  $r$  im Ursprung platziert. Durch Induktion über  $h$  wird bewiesen, dass mindestens sieben der vier *extremen Eckpunkte*  $(\pm \frac{3^h-1}{2}, \pm \frac{3^h-1}{2})$  und vier der *extremen Zentralpunkte*  $(\pm \frac{3^h-1}{2}, 0)$  sowie  $(0, \pm \frac{3^h-1}{2})$  durch einen Knoten oder einer Kante von  $T(h)$  belegt sind. Es ist zu beachten, dass einer der extremen Eckpunkte durch die eingehende Kante von  $T(h)$  belegt wird. Die Aussage gilt für  $h = 0$  und  $h = 1$ .

Seien  $T_1, \dots, T_7$  sieben vollständige Bäume der Kinder von  $r$  mit Höhe  $h - 1$  und mit den Wurzeln  $r_1, \dots, r_7$ . O. B. d. A. zeigen die Kanten  $(r, r_1), \dots, (r, r_7)$  in beliebige Richtungen außer nach Nordwesten. Das erlaubt die Nummerierung der  $T_i$ 's gegen den Uhrzeigersinn beginnend im Westen.

Angenommen der Gitterpunkt  $p = (\frac{3^h-1}{2}, 0)$  ist nicht durch  $\Gamma(T(h))$  belegt. Daher belegt der Unterbaum  $T_5$  mit der nach Osten zeigenden eingehenden Kante  $(r, r_5)$  den Gitterpunkt  $p$  nicht. Unter Verwendung der Induktionsannahme hat die Zeichnung von  $T_5$  mindestens die Seitenlängen  $3^{h-1} - 1$ . Der eine extreme Eckpunkt von  $T_5$  belegt die Diagonale von  $r$  in Richtung Nordosten. Deshalb ist es nicht möglich die Diagonale für  $T_6$  zu verwenden. Dies kann analog für  $T_1, T_3$  und  $T_7$  mit symmetrischen Argumenten gezeigt werden.

Es bleibt noch zu zeigen, dass die extremen Eckpunkte belegt sind. Angenommen der Gitterpunkt  $q = (\frac{3^h-1}{2}, -\frac{3^h-1}{2})$  ist nicht durch die Zeichnung  $\Gamma(T(h))$  belegt.



**Abbildung 3.28.**  $OLP_8$ -Zeichnung eines vollständigen 5-nären Baumes (Kor. 3.4)

Dann können  $T_4$  und die nach Südosten zeigende Kante  $(r, r_4)$  den Gitterpunkt  $q$  nicht belegen. Somit sind die Extrempunkte von  $T_4$  höchstens Richtung Norden und/oder Richtung Westen verschoben. O. B. d. A. sind diese Extrempunkte um eins in Richtung Norden (die beiden anderen Möglichkeiten analog) verschoben. Da  $T_4$  und  $T_5$  mindestens die Seitenlänge  $3^{h-1} - 1$  haben und weil sich  $T_4$  und  $T_5$  nicht überlagern, überlagert  $T_5$  die Diagonale von  $r$  in Richtung Nordosten. Deshalb ist es nicht möglich die diagonale Richtung für  $T_6$  zu verwenden. Das Gleiche kann für  $T_2$  und  $T_6$  durch symmetrische Argumente gezeigt werden.  $\square$

**Korollar 3.2.** *Es gibt einen Linearzeitalgorithmus zum Zeichnen von vollständigen 7-nären Bäumen mit  $n$  Knoten auf dem 8-er Gitter in der Fläche  $\mathcal{O}(n^{1.130})$  und mit Seitenverhältnis 1.*

Ausgehend von 7-nären vollständigen Bäumen auf dem 8-er Gitter werden im Folgenden 6-näre und 5-näre Bäume betrachtet.

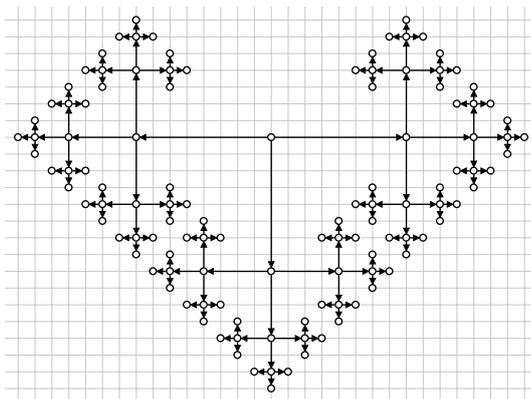
**Korollar 3.3.** *Es gibt einen Linearzeitalgorithmus zum Zeichnen von vollständigen 6-nären Bäumen mit  $n$  Knoten auf dem 8-er Gitter in der Fläche  $\mathcal{O}(n^{\log_6 9}) \subset \mathcal{O}(n^{1.227})$  und mit Seitenverhältnis 1.*

*Beweis.* Der Beweis der oberen Schranke für Zeichnungen von 6-nären Bäumen auf dem 8-er Gitter ist analog zum Beweis der oberen Schranke aus Satz 3.8, wobei für  $h = \log_6 n$  eingesetzt wird.  $\square$

**Korollar 3.4.** *Es gibt einen Linearzeitalgorithmus zum Erstellen einer lokal-uniformen Pattern-Zeichnung von vollständigen 5-nären Bäumen mit  $n$  Knoten auf dem 8-er Gitter in der Fläche  $\mathcal{O}(n^{1.366})$ .*

*Beweis.* Der Beweis kann analog zum Beweis Satz 3.1 ausgeführt werden, siehe dazu Abbildungen 3.28(a) und 3.28(b).  $\square$

Ein Beispiel für die Zeichnung auf dem 8-er Gitter ist in Abbildung 3.28(c) gezeigt. Im nächsten Theorem wird ebenfalls eine obere Schranke bezüglich der Fläche für  $OLP_8$ -Zeichnungen von 4-nären Bäumen betrachtet.



**Abbildung 3.29.** Lokal uniforme Zeichnung eines vollständigen Ternärbaumes auf dem 4-er Gitter

**Korollar 3.5.** *Es gibt einen Linearzeitalgorithmus zum Erstellen von lokal-uniformen Pattern-Zeichnungen von vollständigen 4-nären Bäumen mit  $n$  Knoten auf dem 8-er Gitter in der Fläche  $\mathcal{O}(n^{1.585})$ .*

*Beweis.* Sei  $T$  ein vollständiger 4-närer Baum der Höhe  $h$ . Die Konstruktion wird ebenfalls unter Beachtung der Pattern und der lokalen Uniformität ausgeführt. Die weiteren Schritte sind analog zum Beweis von Kor. 3.4, so dass sich eine Fläche in  $\mathcal{O}(n^{\log_4 9}) \subset \mathcal{O}(n^{1.585})$  ergibt.  $\square$

**Satz 3.9.** *Eine lokal-uniforme Zeichnung eines vollständigen Ternärbaumes mit  $n$  Knoten auf dem 4-er Gitter kann in der Fläche  $\mathcal{O}(n^{1.262})$  planar gezeichnet werden.*

*Beweis.* Sei  $T = (V, E)$  ein vollständiger Ternärbaum der Höhe  $h$ . Ein Ternärbaum mit Höhe  $h = 1$  kann o. B. d. A. mit Höhe  $H = 1$  und Breite  $W = 2$  gezeichnet werden. Im Konstruktionsschritt  $i \rightarrow i + 1$  werden drei Ternärbäume  $T_1, T_2, T_3$  der Höhe  $i$  mit den Wurzeln  $v_1, v_2, v_3$  zu einem Ternärbaum  $T$  der Höhe  $i + 1$  mit der Wurzel  $v$  zusammengefügt. Dabei haben o. B. d. A. die eingehenden von  $v_1$  und  $v_2$  entgegengesetzte horizontale Richtungen und die eingehende Kante von  $v_3$  eine vertikale Richtung, siehe Abbildung 3.29. Um Planarität zu garantieren, verdoppelt sich die Kantenlänge im Konstruktionsschritt  $i \rightarrow i + 1$  von  $(v, v_i)$  im Vergleich zu den Kindern von  $v_1, v_2$  oder  $v_3$ . Folglich ergibt sich mit  $h = \log_3 n$  in diesem Fall eine Breite von  $\sum_{j=0}^n 2^h = 2^{h+1} - 1 \in \Omega(2^h)$  und damit eine Breite in  $\Omega(2^{\log_3 n})$ . Analog kann dies für die Höhe der Zeichnung ausgeführt werden.  $\square$

Für vollständige Bäume ohne Knotenbeschriftung ist es unerheblich, ob sie geordnet oder ungeordnet sind, wodurch diese Tatsache in diesem Abschnitt außer Acht gelassen werden kann. Im nächsten Abschnitt werden beliebige geordnete Bäume auf den verschiedenen Gittern diskutiert.

## 3.2.2 Geordnete Bäume

Nach den Abschätzungen für vollständige  $d$ -näre Bäume werden in diesem Abschnitt beliebige geordnete  $d$ -näre Bäume diskutiert. In hierarchischen Zeichnungen von geordneten Binärbäumen auf ganzzahligen Koordinaten mit den Ästhetikkriterien  $AK1 - AK4$  ist die untere Schranke für die Fläche in  $\Omega(n^2)$ , siehe dazu auch das Beispiel in Abbildung 3.7. Die lineare Höhe der Zeichnung ist bedingt durch das Ästhetikkriterium  $AK1$  (Definition 3.2) und die lineare Breite aufgrund von  $AK2$  (Definition 3.3). Für die  $k$ -Gitter können ebenfalls verschiedene Ästhetikkriterien gefordert werden. Diese sind zusätzlich zu Planarität und Zeichnen auf ganzzahligen Koordinaten der Knoten und Knicke beispielsweise die Ordnungserhaltung, das Zeichnen nach den Pattern (Definition 3.7) und die lokale Uniformität (Definition 3.9). Des Weiteren wird oft Subgraph-Isomorphie verlangt, um gleiche Strukturen in Zeichnungen schneller und einfacher erkennen zu können. Ebenfalls ist es sinnvoll die Rechteck Unterbaum Separation zu fordern, um übersichtlichere Zeichnungen zu erhalten. Letztere Kriterien widersprechen sich jedoch in den meisten Fällen. Dies ist schon in hierarchischen Zeichnungen festzustellen.

Im Fortgang werden ordnungserhaltende Zeichnungen auf dem  $k$ -Gitter als  $O_k$ -Zeichnungen bezeichnet (Abbildung 3.18(a)). Sind diese zusätzlich noch lokal uniform, so heißen sie  $OL_k$ -Zeichnungen (Abbildung 3.18(b)). Für ordnungserhaltende Pattern-Zeichnungen wird die Abkürzung  $OP_k$ -Zeichnungen verwendet, siehe Beispiel in Abbildung 3.18(c). Ist eine ordnungserhaltende Zeichnung lokal uniform und nach den Pattern gezeichnet, so wird sie als  $OLP_k$ -Zeichnung betitelt (Abbildung 3.18(d)).

Zuerst wird eine Übersicht für untere Schranken für geordnete Zeichnungen von  $d$ -nären Bäumen ( $d < k$  und  $k \in \{4, 6, 8\}$ ) in Tabelle 3.3 vorgestellt, deren Ergebnisse nachfolgend verifiziert und erläutert werden. Für ordnungserhaltende Zeichnungen von Bäumen bis zum Grad vier hat Frati eine feste untere und obere Schranke für die Fläche angegeben, die in  $\Theta(n^2)$  liegt [Fra07]. Des Weiteren wird die Komplexität, die minimale Fläche einer Zeichnung unter Beachtung bestimmter Ästhetikkriterien zu berechnen, diskutiert.

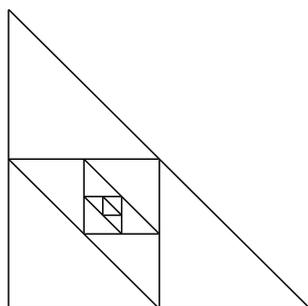
### 3.2.2.1 Flächenschranken

In den folgenden Abschnitten werden zuerst für  $k \in \{4, 6, 8\}$   $O_k$ -Zeichnungen, danach  $OL_k$ -,  $OP_k$ - und zuletzt  $OLP_k$ -Zeichnungen untersucht. Dabei wird auf untere und obere Schranken eingegangen. Die in diesem Abschnitt behandelten unteren Schranken für ordnungserhaltende Zeichnungen auf  $k$ -Gittern mit  $k \in \{4, 6, 8\}$  sind in Tabelle 3.3 zusammengefasst.

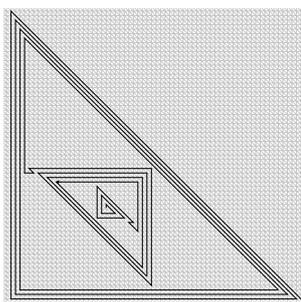
**$O_k$ -Zeichnungen** Zu Beginn dieses Abschnittes werden untere Schranken für ordnungserhaltende Zeichnungen geordneter Bäume auf den verschiedenen Gittern gezeigt, d. h. für  $O_k$ -Zeichnungen mit  $k \in \{4, 6, 8\}$ .

	$k = 4$	$k = 6$	$k = 8$
$O_k$	$\Theta(n^2)$ , [Fra07]	$\Omega(4^n)$ , Satz 3.10	$\Omega(4^n)$ , Satz 3.10
$OP_k$	$\Theta(n^2)$ , Kor. 3.6	$\Omega(4^n)$ , Kor. 3.7	$\Omega(4^n)$ , Kor. 3.7
$OL_k$	$\Omega(2^{\sqrt{n}})$ , Verm. 3.1	$\Omega(4^n)$	$\Omega(4^n)$
$OLP_k$	$\Omega(2^{\sqrt{n}})$ , Verm. 3.1	$\Omega(4^n)$	$\Omega(4^n)$

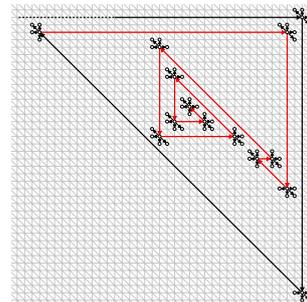
**Tabelle 3.3.** Untere Schranken für die Fläche von ordnungserhaltende Zeichnungen auf  $k$ -Gittern mit  $k \in \{4, 6, 8\}$



(a) Beispiel Dreieckszerlegung



(b) Skizze Beweis Satz 3.10



(c) Ausschnitt aus Abbildung 3.30(b)

**Abbildung 3.30.** Untere Schranke für  $O_k$ -Zeichnungen für  $k \in \{6, 8\}$

Um die untere Schranke für ordnungserhaltende Zeichnungen von 5-nären (7-nären) Bäumen auf dem 6-er (8-er) Gitter zu bestimmen wird Folgendes festgestellt. Gegeben sei ein rechtwinkliges Dreieck  $\Delta_n$  mit den ganzzahligen Koordinaten  $(0, 0)$ ,  $(2^n, 0)$ ,  $(0, 2^n)$  der Eckpunkte. Solange  $i > 1$  wird im Zerlegungsschritt von  $i \rightarrow i - 1$  aus  $\Delta_i$  das Dreieck  $\Delta_{i-1}$  mit den Koordinaten der Mittelpunkte der Seiten von  $\Delta_i$  gebildet. Aufgrund der ursprünglichen durch zwei teilbaren Koordinaten von  $\Delta_i$  sind die Koordinaten von  $\Delta_{i-1}$  wieder ganzzahlig. Diese Konstruktion wird *Dreieckszerlegung* genannt. Ein Beispiel der Dreieckszerlegung ist in Abbildung 3.30(a) gezeigt.

**Lemma 3.2.** Sei  $\Delta_n$  ein rechtwinkliges gleichschenkliges Dreieck mit den Koordinaten der Eckpunkte  $(0, 0)$ ,  $(2^n, 0)$ ,  $(0, 2^n)$ . Das Verhältnis der Flächen  $\frac{A(\Delta_n)}{A(\Delta_1)}$ , erzeugt durch die Dreieckszerlegung ist in  $\Omega(4^n)$ .

*Beweis.* Sei  $\Delta_n$  ein rechtwinkliges gleichschenkliges Dreieck mit den Koordinaten  $(0, 0)$ ,  $(0, 2^n)$  und  $(2^n, 0)$ . Der Konstruktionsschritt  $i \rightarrow i - 1$  erzeugt wieder ein rechtwinkliges gleichschenkliges Dreieck  $\Delta_{i-1}$  mit Seitenlänge der Katheten von  $2^{i-1}$ . Im Konstruktionsschritt halbiert sich die Seitenlänge und die Fläche ist  $A(\Delta_{i-1}) = \frac{1}{4}A(\Delta_i)$ . Da dieser Schritt  $n$ -mal ausgeführt wird bis das rechtwinklige Dreieck  $\Delta_1$

entsteht, haben die Katheten in  $\Delta_1$  die Länge 1. Aufgrund der Viertelung der Fläche in jedem Schritt ist das Verhältnis von  $\frac{A(\Delta_n)}{A(\Delta_1)}$  in  $\Omega(4^n)$ .  $\square$

**Satz 3.10.** *Es gibt einen geordneten  $d$ -nären Baum  $T$ , dessen ordnungserhaltende Zeichnung auf dem  $k$ -Gitter mit  $k \in \{6, 8\}$  und  $d < k$  eine Fläche in  $\Omega(4^n)$  benötigt.*

*Beweis.* Aufgrund der Ordnungserhaltung in Zeichnungen von  $d$ -nären Bäumen für  $k \in \{6, 8\}$  und  $d < k$  kann ein Baum  $T = (V, E)$  konstruiert werden, der sich einwickelt. Dazu wird ein Pfad  $(v_0, \dots, v_n)$  der Länge  $n$  definiert und die Knoten  $v_0, \dots, v_n$  werden *Pfadknoten* genannt. Jeder Pfadknoten  $v_i \in \{v_0, \dots, v_{n-1}\}$  besitzt ein Kind  $v_{i+1}$ . Sei  $c = 6$  eine Konstante. Jedem Knoten  $v_i$  werden  $k - 2$  weitere Kinder hinzugefügt. Ein Pfadknoten  $v_i$  mit der Tiefe  $t(v_i)$  ist das linkeste Kind, wenn  $\lfloor \frac{t(v_i)}{c} \rfloor \bmod 2 = 0$  gilt, ansonsten ist es das rechteste Kind. Durch die alternierenden Kindbeziehungen der Pfadknoten wird erzwungen, dass sich ein Teilpfad der Länge  $c$  einmal nach links und der darauffolgende nach rechts eindreht. Da die Kantenlänge mindestens eins ist, ist aufgrund von Lemma 3.2 die Fläche in  $\Omega(4^n)$ .  $\square$

Eine Beispielskizze zum Beweis von Satz 3.10 ist in Abbildung 3.30(b) und ein konkreter Ausschnitt des Baumes in Abbildung 3.30(c) gezeigt. Im Gegensatz zu ordnungserhaltenden Zeichnungen von geordneten Ternärbäumen auf dem 4-er Gitter, die mindestens quadratische Fläche brauchen, haben 5-näre und 7-näre Bäume auf dem 6-er und 8-er Gitter eine exponentielle untere Schranke. Auf dem 4-er Gitter würde dieser Ansatz die bekannte untere Schranke  $\Omega(n^2)$  bestätigen.

**Satz 3.11.** *Jeder  $d$ -näre Baum mit  $n$  Knoten kann auf dem  $k$ -Gitter mit  $k \in \{6, 8\}$  und  $d < k$  in der Fläche  $\mathcal{O}(9^n)$  ordnungserhaltend gezeichnet werden.*

*Beweis.* Gegeben sei ein  $d$ -närer Baum  $T$  mit  $k \in \{6, 8\}$  und  $d < k$ . Nach der Konstruktion aus dem Beweis von Satz 3.1 bzw. Satz 3.8 wächst die Fläche von  $h \rightarrow h+1$  um das neunfache, so dass sich ein  $d$ -närer Baum durch die Methode „vollständig Zeichnen und Löschen der nicht existenten Knoten“ in der Fläche  $\mathcal{O}(9^{h(T)})$  und somit in  $\mathcal{O}(9^n)$  zeichnen lässt.  $\square$

Bis jetzt wurden die oberen und unteren Schranken für  $O_k$ -Zeichnungen mit  $k \in \{4, 6, 8\}$  diskutiert, was nachfolgend zu Zeichnungen mit den lokal uniformen und/oder Pattern-Zeichnungen führt.

**$OP_k$ -Zeichnungen** In ordnungserhaltenden Pattern-Zeichnungen, den  $OP_k$ -Zeichnungen auf  $k$ -Gittern, werden die Kantenrichtungen durch die Pattern vorherbestimmt. Die Konstruktion von  $O_k$ -Zeichnungen im Beweis von Satz 3.10 erzeugt bereits  $OP_k$ -Zeichnungen. Somit gelten nach [Fra07] und Satz 3.10 die unteren Schranken ebenfalls für  $OP_k$ -Zeichnungen.

**Korollar 3.6.** *Es gibt einen geordneten Ternärbaum  $T$ , dessen  $OP_4$ -Zeichnung auf dem 4-er Gitter eine Fläche in  $\Omega(n^2)$  benötigt.*

**Korollar 3.7.** *Es gibt einen geordneten  $d$ -nären Baum  $T$ , dessen  $OP_k$ -Zeichnung auf dem  $k$ -Gitter mit  $k \in \{6, 8\}$  und  $d < k$  eine Fläche in  $\Omega(4^n)$  benötigt.*



	$k = 4$	$k = 6$	$k = 8$	Satz
$O_k$	$\mathcal{NP}$	$\mathcal{NP}$	$\mathcal{NP}$	3.13
$OL_k$	$\mathcal{NP}$			3.14
$OP_k$	$\mathcal{NP}$	$\mathcal{NP}$	$\mathcal{NP}$	3.15
$OLP_k$	$\mathcal{NP}$	$\mathcal{NP}$	$\mathcal{NP}$	3.16

**Tabelle 3.4.** Minimale Fläche für ordnungserhaltende Zeichnungen

### 3.2.2.2 Minimale Fläche

In diesem Abschnitt wird die Komplexität für das Bestimmen von ordnungserhaltenden Zeichnungen mit minimaler Fläche bestimmt. Dabei kann in den meisten Fällen die  $\mathcal{NP}$ -Härte festgestellt werden. Das trifft für  $O_k$ -,  $OP_k$ - und  $OLP_k$ -Zeichnungen zu, jedoch für ordnungserhaltende lokal uniforme Zeichnungen nur für  $k = 4$ . Eine Übersicht der  $\mathcal{NP}$ -harten Probleme auf den verschiedenen  $k$ -Gittern ist in Tabelle 3.4 zu sehen. Als Ästhetikkriterien werden die Ganzzahligkeit, Geradlinigkeit der Kanten, bis auf Verschiebung und Rotation identische Zeichnungen für isomorphe Unterbäume und Planarität gefordert.

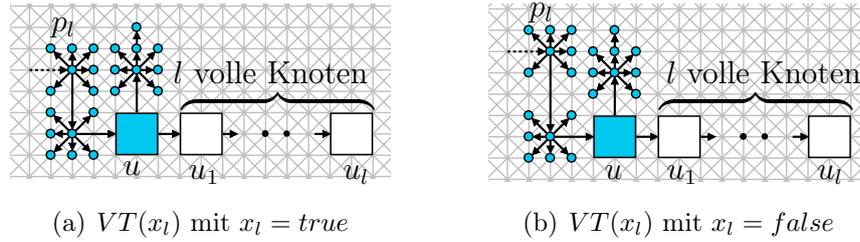
**$O_k$ -Zeichnungen** Das Problem des Zeichnens auf minimale Fläche ist in  $\mathcal{NP}$ , da durch „guess and check“ eine Zeichnung auf minimaler Fläche geraten werden und dann verifiziert werden kann, ob die Zeichnung die Ästhetikkriterien erfüllt. Für  $O_k$ -Zeichnungen wird im Folgenden gezeigt, dass es  $\mathcal{NP}$ -hart ist, die minimale Fläche unter Einhaltung der gegebenen Ästhetikkriterien zu bestimmen. Um die  $\mathcal{NP}$ -Härte zu zeigen wird erst die Definition eines bekannten  $\mathcal{NP}$ -vollständigen Problems angegeben.

**Definition 3.10 (3-SAT).** [GJ90] Gegeben sei eine Menge  $X = x_1, \dots, x_n$  von Variablen und eine Menge  $F = F_1, \dots, F_c$  von Klauseln. Jede Klausel besteht aus drei disjunktiv verknüpften Literalen  $F_i = (y_{i,1} \vee y_{i,2} \vee y_{i,3})$ . Die konjunktive Verknüpfung der Klauseln ergibt dann einen booleschen Ausdruck  $E = F_1 \wedge F_2 \wedge \dots \wedge F_c$  in 3-KNF. Gefragt ist dann, ob  $E$  erfüllbar ist, d. h. existiert für jede Variable  $x_j \in X$  eine Belegung  $\beta(x_j) \in \{true, false\}$ , so dass  $E$  true ist?

**Satz 3.12.** 3-SAT ist  $\mathcal{NP}$ -vollständig [GJ90].

Es wird vom  $\mathcal{NP}$ -vollständigen Problem 3-SAT reduziert, indem die Eingabe eines booleschen Ausdruck  $E$  in konjunktiver Normalform mit höchstens drei Literalen pro Klausel ist. Zu entscheiden, ob es eine Belegung  $\beta$  für die Variablen gibt, die den Ausdruck  $E$  erfüllt, ist nach Satz 3.12  $\mathcal{NP}$ -vollständig.

Die Reduktion konstruiert einen  $(k - 1)$ -ären Baum aus dem gegebenen booleschen Ausdruck in polynomialer Zeit. Die Konstruktion hat eine gewisse Ähnlichkeit



**Abbildung 3.32.** Zeichnungen des Variablenbaumes  $VT(x_l)$

zur Konstruktion von Supowit und Reingold im hierarchischen Fall [SR82,ABJT04]. Sei  $E$  ein beliebiger boolescher Ausdruck in  $\beta$ -KNF mit  $c$  Klauseln  $F_1, \dots, F_c$  und  $n$  Variablen  $x_1, \dots, x_n$ . Jede Klausel  $F_i$  mit  $i \in \{1, \dots, c\}$  enthält drei Literale, wobei  $F_i = (y_{i,1} \vee y_{i,2} \vee y_{i,3})$ . Ein Literal  $y_{i,j}$  mit  $j \in \{1, 2, 3\}$  in der  $i$ -ten Klausel ist entweder eine Variable  $x_l$  oder ihre Negation  $\overline{x_l}$ .

Im Folgenden wird die Reduktion für den Fall  $k = 8$  im Detail erläutert, indem aus dem gegebenen booleschen Ausdruck ein 7-närer Baum konstruiert und dessen Einbettung auf dem 8-Gitter beschrieben wird.

Um die Konstruktion vereinfacht beschreiben zu können, werden Knoten mit sieben Kindern als *volle Knoten* bezeichnet. In den Darstellungen werden die vollen Knoten und deren Blätter mit quadratischen Boxen gezeichnet, sogenannte *Boxknoten*, siehe Abbildungen 3.32(a) und 3.32(b). Die Beschriftungen eines Boxknotens oder eines vollen Knotens und dessen Blätter identifizieren immer den zentralen Knoten darin. Ein Pfad von vollen Knoten wird *voller Pfad* genannt.

**Variablenbaum** Für eine Variable  $x_l$  mit  $l \in \{1, \dots, n\}$  des booleschen Ausdrucks  $E$  wird ein eindeutiger 7-närer Variablenbaum  $VT(x_l)$  definiert, vgl. Abbildung 3.32. Es ist der induzierte Unterbaum des Wurzelknotens  $p_l$  mit vollem Pfad  $(p_l, \dots, u)$ . Der volle Pfad  $(u_1, \dots, u_l)$  mit der Länge  $l$  abhängig vom Index der Variablen  $x_l$  wird an  $u$  gehängt. Über  $u$  wird ein weiterer voller Knoten  $h$  angefügt. An  $h$  wird oberhalb ein einzelner Knoten  $z$  angebracht. Beachte, dass der gleiche Baum aus Abbildung 3.32(a) in Abbildung 3.32(b) dargestellt ist, mit dem Unterschied, dass die vertikal ausgehende Kante von  $p_l$  die Länge vier statt drei hat.

**Literalbaum** Das  $j$ -te Literal  $y_{i,j}$  in der  $i$ -ten Klausel ist entweder die nicht negierte Variable  $x_l$  oder die negierte Form  $\overline{x_l}$ . Zuerst wird ein *Literalbaum*  $LT(y_{i,j})$  definiert, der einen Variablenbaum  $VT(x_l)$  enthält. Das Skelett eines Literalbaumes (vgl. Abbildung 3.33) ist der volle Pfad  $(b_{i,j}, \dots, f_{n+i+1})$ . An  $f_{l+1}$  wird ein Pfad von drei vollen Knoten  $P_{grenze}$  in vertikaler Richtung angefügt. Des Weiteren wird diese Konstruktion oben ebenfalls in ähnlicher Weise ohne  $P_{grenze}$  in Form eines vollen Pfades von  $b_{i,j}$  nach  $g_{n+i+1}$  hinzugefügt. Wie man später sehen kann, stellt  $P_{grenze}$  sicher, dass die Breite des Variablenbaumes beschränkt ist. Oberhalb des Kindes des letzten Knotens  $f_{n+i+1}$  wird ein Pfad  $P_f$  von  $j$  einzelnen Knoten gehängt (analog

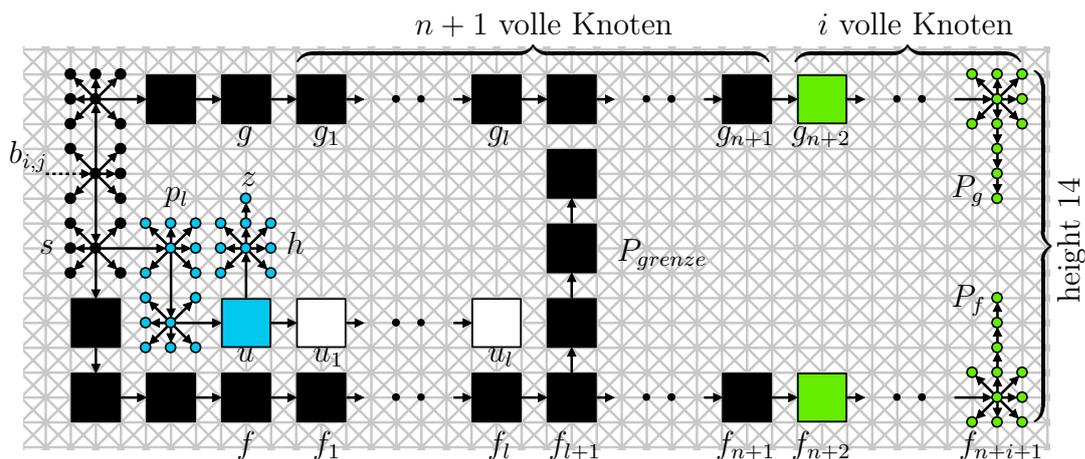


Abbildung 3.33. Literalbaum  $LT(y_{i,3})$  mit der Variable  $y_{i,3} = x_l$

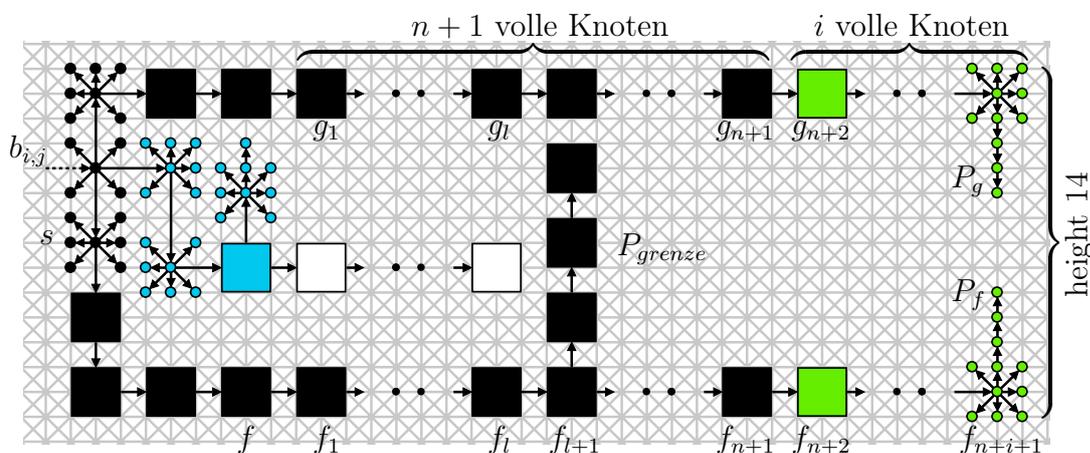


Abbildung 3.34. Literalbaum  $LT(y_{i,j})$  von  $y_{i,j}$  mit  $j = 3$ ,  $y_{i,j} = \bar{x}_l$

$P_g$  unterhalb von  $g_{n+i+1}$ ), siehe Abbildung 3.33. Aufgrund dieser Konstruktion sind alle diese Unterbäume in den verschiedenen Literalbäumen nicht isomorph.

Wenn die Variable  $y_{i,j} = x_l$  nicht negiert ist, dann wird die Wurzel  $p_l$  des Variablenbaumes  $VT(x_l)$  an den Knoten  $s$  angefügt, vgl. Abbildung 3.33. Für den Fall, dass  $y_{i,j} = \bar{x}_l$  negiert ist, wird  $p_l$  an  $b_{i,j}$  angehängt, siehe Abbildung 3.34.

**Höhe und Breite eines Literalbaumes:** Sei die Richtung der eingehenden Kante von  $b_{i,j}$  von links nach rechts. Abbildung 3.33 zeigt eine Zeichnung von  $LT(y_{i,j})$  mit  $y_{i,j} = x_l$  mit minimaler Höhe 14. Beachte, dass alle Kantenrichtungen durch die Ordnungserhaltung bis auf die Kanten von  $P_f$ ,  $P_g$  und die eingehende Kante von  $z$  durch die eingehende Kante der Wurzel  $b_{i,j}$  festgelegt sind. Wenn alle vertikalen Kanten die minimale Länge besitzen, dann hat die Zeichnung des Literalbaumes

minimale Höhe 14. Ansonsten wäre die Zeichnung des Literalbaumes höher.

Die minimale Höhe der Zeichnung eines Literalbaumes  $LT(y_{i,j})$  mit negierter Variable  $y_{i,j} = \bar{x}_l$  ist ebenfalls 14, wie in Abbildung 3.34 zu sehen ist. Es ist zu beachten, dass sich die Höhe des Literalbaumes aufgrund von  $z$  um eins erhöht, wenn der ausgehende Kante von  $p_l$  die Länge drei zugewiesen wird.

In beiden Fällen  $y_{i,j} = x_l$  und  $y_{i,j} = \bar{x}_l$  hat die Zeichnung des Literalbaumes  $LT(y_{i,j})$  aufgrund der Pfade  $(f, \dots, f_{n+i+1})$  und  $(g, \dots, g_{n+i+1})$  eine minimale Breite von  $8 + 3(n + i + 1)$ .

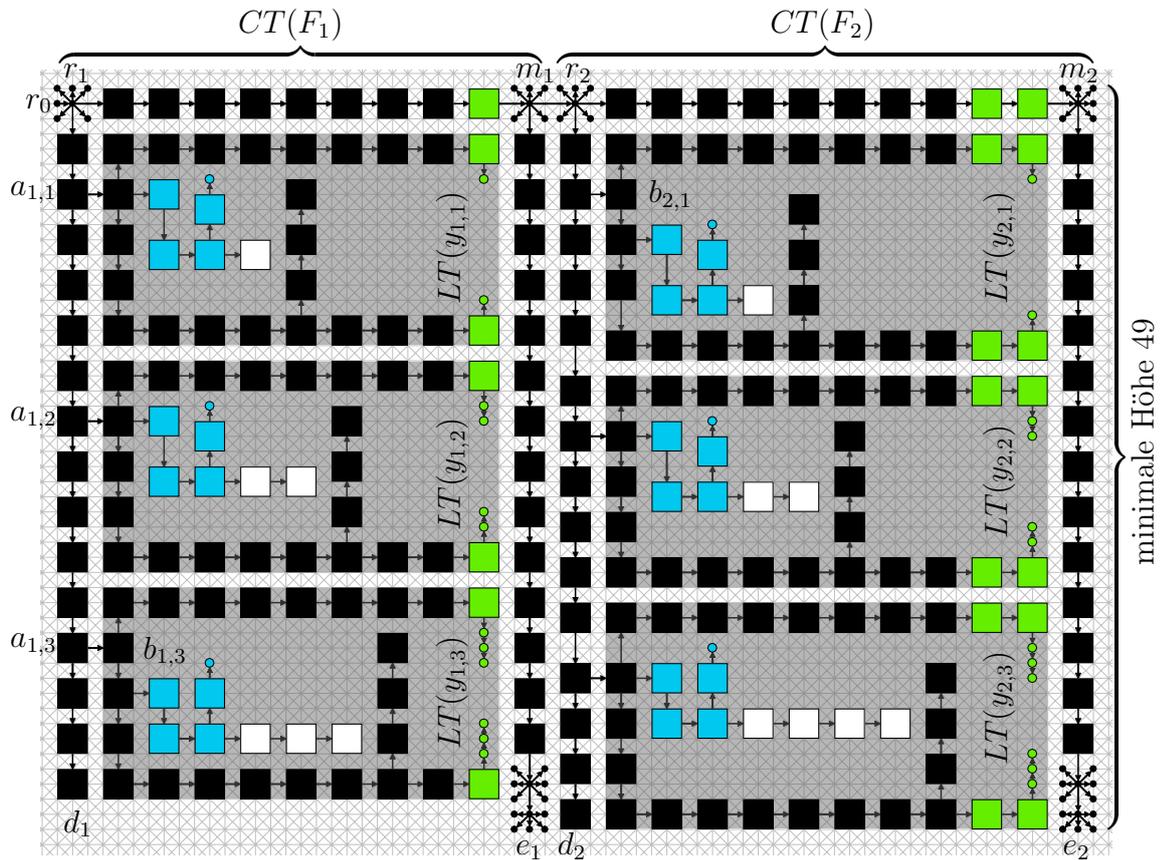
**Klauselbaum** Für eine Klausel  $F_i = (y_{i,1} \vee y_{i,2} \vee y_{i,3})$  wird ein 7-närer Klauselbaum  $CT(F_i)$  mit der Wurzel  $r_i$  definiert. Jeder Klauselbaum  $CT(F_i)$  hat die vollen Pfade  $(r_i, \dots, m_i)$ ,  $(m_i, \dots, e_i)$  und  $(r_i, \dots, d_i)$ , vgl. Abbildung 3.35. Der Klauselbaum  $CT(F_i)$  enthält drei Literalbäume  $LT(y_{i,1})$ ,  $LT(y_{i,2})$  und  $LT(y_{i,3})$  mit den Wurzeln  $b_{i,1}$ ,  $b_{i,2}$  und  $b_{i,3}$ , die an die entsprechenden Knoten  $a_{i,1}$ ,  $a_{i,2}$  und  $a_{i,3}$  angefügt werden. Ein Beispiel ist in Abbildung 3.35 zu sehen. Dabei verbindet die Kante  $(a_{1,3}, b_{1,3})$  den ersten Klauselbaum  $F_1$  mit seinem dritten Literalbaum  $LT(y_{1,3})$ .

**Höhe und Breite eines Klauselbaumes:** Sei die Richtung der eingehenden Kante der Wurzel  $r_i$  des Klauselbaumes  $F_i$  von links nach rechts. Wie oben erwähnt sind aufgrund der Ordnungserhaltung alle wichtigen Kantenrichtungen festgelegt. Jeder Klauselbaum  $CT(F_i)$  hat einen vertikalen vollen Pfad  $(m_i, \dots, e_i)$ , der die minimale Höhe mit 49 festlegt, vgl. Abbildung 3.35. Wenn jeder Literalbaum eines Klauselbaumes mindestens Höhe 15 hat, dann hat der Klauselbaum mindestens Höhe  $2 + 1 + 15 + 1 + 15 + 1 + 15 = 50$ . Wenn aber mindestens ein Literalbaum mit Höhe 14 gezeichnet wird und die restlichen zwei höchstens die Höhe 15 haben, dann kann der Klauselbaum mit Höhe 49 gezeichnet werden.

Der oberste horizontale Pfad  $(r_i, \dots, m_i)$  von vollen Knoten bestimmt die kleinstmögliche Breite der Zeichnung abhängig von der aktuellen Klausel  $F_i$  und der Anzahl der Variablen  $n$  mit  $14 + 3(n + i + 1)$ .

**Baum  $T(E)$**  Für einen booleschen Ausdruck  $E = F_1 \wedge \dots \wedge F_c$  wird ein 7-närer Baum  $T(E)$  mit den Klauselbäumen  $CT(F_1), \dots, CT(F_c)$  definiert. Die Wurzel  $r_0$  von  $T(E)$  ist der Vater der Wurzel  $r_1$  des ersten Klauselbaumes  $F_1$ . Für  $i \in \{1, \dots, c - 1\}$  ist jeder Klauselbaum  $CT(F_i)$  mit Klauselbaum  $CT(F_{i+1})$  durch die Kante  $(m_i, r_{i+1})$  verbunden. An den letzten Klauselbaum  $CT(F_r)$  wird ein Blatt  $m_{c+1}$  an den Zentrums-knoten  $m_c$  angefügt, dessen eingehende Kante nach rechts zeigt, siehe Abbildung 3.35.

**Minimale Höhe und Breite des Baumes  $T(E)$ :** Angenommen, die Kante  $(r_0, r_1)$  zeigt nach rechts, dann erzwingt die beschriebene Konstruktion wegen der Ordnungserhaltung einen horizontalen vollen Pfad  $(r_1, \dots, m_c)$ , der die minimale Breite mit  $\sum_{i=1}^c (14 + 3(n + i + 1)) + (c - 1) = 1.5c^2 + (3n + 19.5)c - 1$  zusichert.



**Abbildung 3.35.** 7-närer Baum  $T(E)$  auf dem 8-Gitter für den booleschen Ausdruck  $E = (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_4})$  und Belegung  $\tau(x_1) = \tau(x_2) = false$  und  $\tau(x_3) = \tau(x_4) = true$

Die Höhe der Zeichnung  $\Gamma(T(E))$  ist das Maximum der Höhen der Klauselbäume, weil die Klauselbäume an einer Linie nebeneinander nach oben hin ausgerichtet sind.

**Lemma 3.3.** *Sei  $E$  ein boolescher Ausdruck in 3-KNF mit  $c$  Klauseln und  $n$  Variablen.  $E$  ist genau dann erfüllbar, wenn es eine ordnungserhaltende Zeichnung  $\Gamma(T(E))$  des geordneten 7-nären Baumes  $T(E)$  auf dem 8-Gitter mit höchstens der Fläche  $A = 49 \cdot W$  gibt, wobei  $W = 1.5c^2 + (3n + 19.5)c - 1$ .*

*Beweis.* Sei  $E$  ein beliebiger boolescher Ausdruck in 3-KNF mit den  $n$  Variablen  $x_1, \dots, x_n$  und den  $c$  Klauseln  $F_1, \dots, F_c$ . Daraus wird der entsprechende 7-näre Baum so konstruiert, wie oben beschrieben.

' $\Rightarrow$ ': Sei  $\tau : \{x_1, \dots, x_n\} \rightarrow \{true, false\}$  eine erfüllende Belegung für den booleschen Ausdruck  $E$ . Dann wird eine Zeichnung des Baumes  $\Gamma(T(E))$  auf dem 8-Gitter mit höchstens der Fläche  $A$  wie folgt konstruiert.

Wenn eine Variable  $x_l$  mit  $\tau(x_l) = true$  belegt ist, dann wird der Variablenbaum  $VT(x_l)$  wie in Abbildung 3.33 mit kürzestmöglichen Kanten gezeichnet. Andernfalls werden für  $\tau(x_l) = false$  alle Kanten kürzestmöglich gezeichnet mit Ausnahme der vertikal ausgehenden Kanten von  $p_l$ , die mit der Länge vier gezeichnet wird, siehe die Zeichnung des Variablenbaumes in Abbildung 3.34.

Für jedes erfüllte Literal im booleschen Ausdruck  $E$  wird eine Zeichnung mit der Höhe 14 konstruiert. Wenn ein Literal durch die Belegung nicht erfüllt wird, dann wird mit Höhe 15 konstruiert.

Im Folgenden wird die vollständige Fallunterscheidung von (i) bis (iv) durchgegangen.

- (i) Sei  $y_{i,j}$  das  $j$ -te Literal in der  $i$ -ten Klausel des booleschen Ausdrucks  $E$ . Wenn  $y_{i,j} = x_l$  und zugleich  $x_l = true$ , dann wird der Literalbaum  $LT(y_{i,j})$  so wie in Abbildung 3.33 gezeichnet, wo jede Kante kürzestmöglich gezeichnet ist. Folglich hat dieser Literalbaum der nicht negierten Variable die minimale Höhe 14.
- (ii) Wenn  $y_{i,j} = x_l$  and  $x_l = false$ , dann wird die ausgehende Kante von  $p_l$  in  $VT(x_l)$  mit der Länge vier gezeichnet, so dass eine vertikale Kante des Skeletts des Literalbaumes  $LT(y_{i,j})$  mit Länge vier gezeichnet werden muss. Folglich beträgt die Höhe der Zeichnung des Literalbaumes 15.
- (iii) Für  $y_{i,j} = \bar{x}_l$  und  $x_l = true$  werden im Variablenbaum  $VT(x_l)$  alle Kanten kürzestmöglich gezeichnet. Deshalb benötigt der Literalbaum  $LT(y_{i,j})$  aufgrund von  $z$  mindestens die Höhe 15.
- (iv) Der letzte Fall ist  $y_{i,j} = \bar{x}_l$  and  $x_l = false$ . Hier wird der Literalbaum  $LT(y_{i,j})$  wie in Abbildung 3.34 mit Höhe 14 gezeichnet.

Wenn in jeder Klausel mindestens ein Literal erfüllt ist, dann wird der Literalbaum des entsprechenden Literals mit Höhe 14 gezeichnet. Die restlichen Literalbäume haben höchstens die Höhe 15. Somit ergibt sich nach der obigen Konstruktion für jeden Klauselbaum eine Höhe von 49.

Die beschriebene Konstruktion erstellt Zeichnungen von Klauselbäumen mit Höhe 49 und Breite  $14 + 3(n + i + 1)$ . Dadurch, dass Klauselbäume nebeneinander platziert werden, bleibt die Höhe der Zeichnung des Baumes  $T(E)$  bei 49 und die Breite ist höchstens  $W$ , d.h. als Fläche ergibt sich dadurch  $49 \cdot W$ .

' $\Leftarrow$ ': Sei  $\Gamma'(T(E))$  eine Zeichnung mit höchstens der Fläche  $A = 49 \cdot W$ . Angenommen die ausgehende Kante von  $r_0$  der Zeichnung  $\Gamma'(T(E))$  zeigt nach rechts und bestimmt dadurch alle restlichen wesentlichen Kantenrichtungen der Zeichnung. *Unwesentliche Kanten* sind dabei die eingehende Kante von  $z$  jedes Variablenbaumes und die eingehenden Kanten sowie die Kanten der Pfade  $P_f$  und  $P_g$  der einzelnen Literalbäume. Die restlichen Kanten des Baumes  $T(E)$  sind *wesentlich* und somit ist ihre Kantenrichtung in der Zeichnung festgelegt. Aufgrund der vertikalen und horizontalen Pfade in der Zeichnung  $\Gamma'(T(E))$  hat diese mindestens Höhe 49 und mindestens Breite  $W$ . Unter Beachtung dieser Voraussetzung hat somit die Zeichnung exakt die Höhe 49 und exakt die Breite  $W$ . Damit hat jede Kante des vollen Pfades  $(r_1, \dots, m_c)$  die kleinstmögliche Länge drei. Für jede Klausel  $F_i$  wird dadurch der Abstand der vertikalen vollen Pfade  $(r_i, \dots, d_i)$  und  $(m_i, \dots, e_i)$  von  $CT(F_i)$  bestimmt. Daher ergibt sich die Breite für den  $i$ -ten Klauselbaum mit  $14 + 3(n + i + 1)$ .

Im Folgenden wird aus der Zeichnung  $\Gamma'(T(E))$  eine erfüllende Belegung  $\tau'(E)$  konstruiert. Wie schon erwähnt, muss in jedem Klauselbaum  $CT(F_i)$  mindestens ein Literalbaum  $LT(y_{i,j})$  die Höhe 14 haben. Wäre das nicht der Fall, dann wäre die Höhe der Zeichnung  $\Gamma'(T(E))$  größer als 49. Die Variable  $x_l$  dieses Literalbaumes  $LT(y_{i,j})$  mit Höhe 14 wird so belegt, dass das Literal  $y_{i,j}$  erfüllt ist. Das ist wohldefiniert, weil es aufgrund der Isomorphie von Unterbäumen nicht sein kann, dass Literalbäume von nicht-negierten und negierten Literalen der gleichen Variable  $x_l$  beide die Höhe 14 haben.

Beachte, dass isomorphe Unterbäume bei jedem Vorkommen in  $\Gamma'(T(E))$  aufgrund des geforderten Ästhetikkriteriums identisch gezeichnet sind. Das bedeutet, dass jeder Variablenbaum  $VT(x_i)$  der entsprechenden Variable  $x_i$  bis auf Verschiebung immer gleich gezeichnet ist.

Wenn also ein Literalbaum  $LT(y_{i,j})$  eines nicht negierten Literals  $y_{i,j} = x_l$  mit Höhe 14 gezeichnet ist, dann wird die Belegung auf  $\tau'(x_l) = true$  gesetzt. Für eine Zeichnung eines Literalbaumes  $LT(y_{i,j})$  eines negierten Literals  $x_{i,j} = \bar{x}_l$  mit Höhe 14 hat die ausgehende Kante von  $p_l$  mindestens Länge vier. Damit dieses Literal erfüllt ist, wird die Belegung dieses Literals  $y_{i,j}$  auf  $\tau'(x_l) = false$  gesetzt. Jeder Klauselbaum enthält mindestens einen Literalbaum mit Höhe 14. Somit erfüllt die konstruierte Belegung  $\tau'$  jede Klausel  $F_i$ . Demzufolge ist auch  $E$  erfüllt. Für die eventuell verbleibenden Variablen, denen noch keine Belegung zugeteilt wurde, wird ein zufälliger Wahrheitswert vergeben, da der boolesche Ausdruck  $E$  durch den bereits festgelegten Teil der Belegung erfüllt ist.  $\square$

Ein Beispiel einer Zeichnung eines 7-nären Baumes auf dem 8-Gitter innerhalb der Höhe 49 und der Breite  $W = 1.5 \cdot 2^2 + (3 \cdot 4 + 19.5) \cdot 2 - 1 = 68$  wird in Abbildung 3.35 gezeigt. Der Baum gehört zum booleschen Ausdruck  $E = (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_4)$  mit zwei Klauseln ( $c = 2$ ) und vier Variablen ( $n = 4$ ). Die Variable

$x_1$  ist mit *false* belegt ( $\tau(x_1) = \text{false}$ ) und kommt im Literal  $y_{1,1}$  der ersten Klausel  $F_1$  und im Literal  $y_{2,1}$  der zweiten Klausel  $F_2$  vor. Der dazugehörige Literalbaum  $LT(y_{1,1})$  des erfüllten Literals  $y_{1,1} = \bar{x}_1$  ist mit der Höhe 14 gezeichnet. Der zum nicht erfüllten Literal  $y_{2,1}$  gehörende Literalbaum  $LT(y_{2,1})$  hat die Höhe 15. Alle Literale in der Klausel  $F_1$  sind erfüllt und dementsprechend sind alle Literalbäume mit Höhe 14 gezeichnet. Die minimale Höhe eines Klauselbaumes, z. B.  $F_1$ , ist durch den vertikalen Pfad  $(m_1, \dots, e_1)$  von vollen Knoten begrenzt. Die Klausel  $F_2$  hat genau ein erfülltes Literal  $y_{2,2}$ , so dass entsprechend der dazugehörige Literalbaum  $LT(y_{2,2})$  Höhe 14 hat.

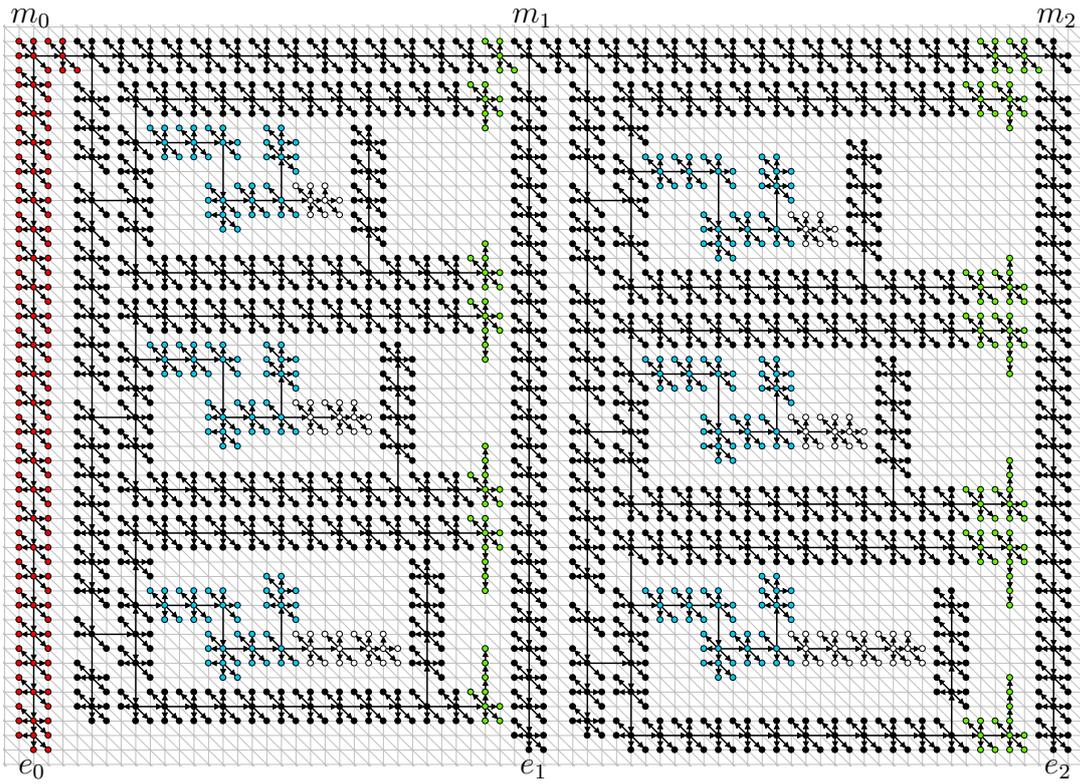
Somit ist gezeigt, dass es für  $k = 8$   $\mathcal{NP}$ -hart ist zu entscheiden ob ein 7-närer Baum auf dem 8-Gitter innerhalb der Fläche  $49 \cdot W$  gezeichnet werden kann. Um die  $\mathcal{NP}$ -Härte für  $k = 6$  zu zeigen muss die Höhe und Breite des Baumes  $T(E)$  angepasst werden. Die Idee der Konstruktion ist bis auf die nachfolgend erläuterten Anpassungen analog zu den Ausführungen auf dem 8-er Gitter.

**Lemma 3.4.** *Sei  $E$  ein boolescher Ausdruck in 3-KNF mit  $c$  Klauseln und  $n$  Variablen.  $E$  ist genau dann erfüllbar, wenn es eine ordnungserhaltende Zeichnung  $\Gamma(T(E))$  des geordneten 5-nären Baumes  $T(E)$  auf dem 6-Gitter mit höchstens der Fläche  $A = 49 \cdot W$  gibt, wobei  $W = c^2 + (29 + n)c + 2$ .*

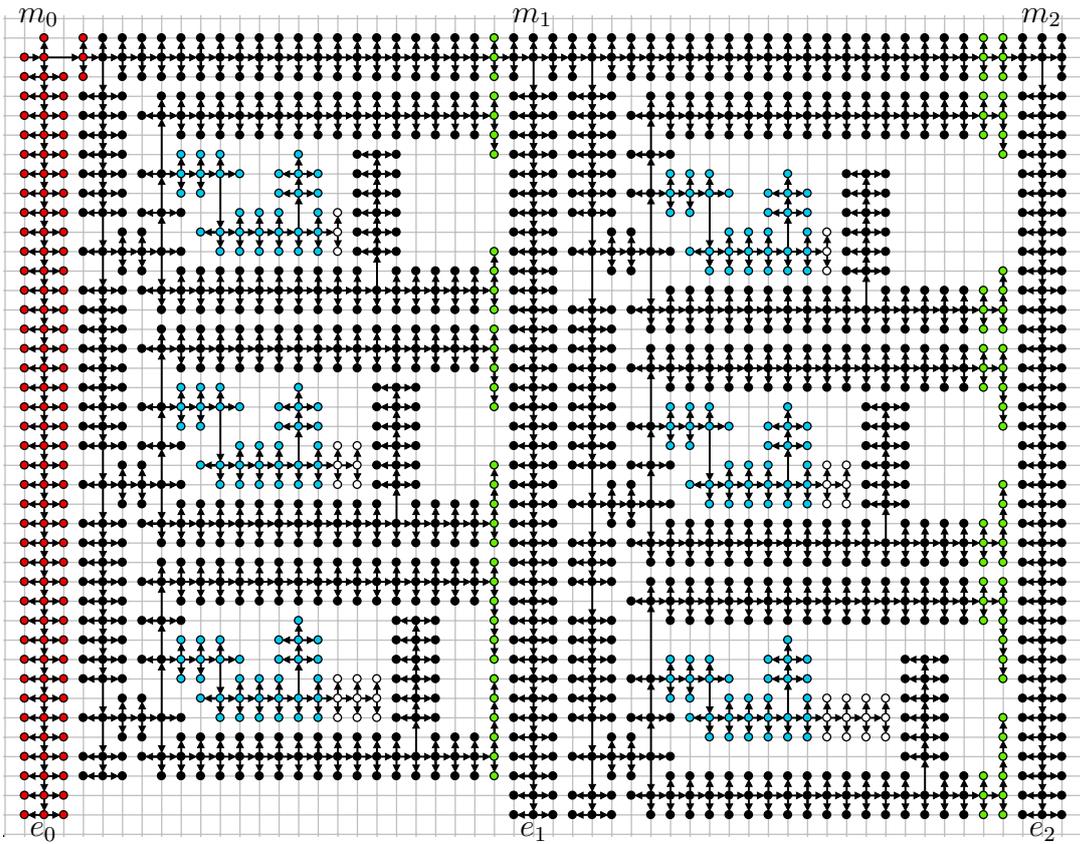
*Beweis.* Es wird vergleichbar zum Beweis von Lemma 3.3 ein 5-närer Baum  $T(E)$  aus einem beliebigen booleschen Ausdruck in 3-KNF mit  $c$  Klauseln und  $n$  Variablen konstruiert. Um den Beweis wie oben durchzuführen, wird ein voller Pfad  $(m_0, \dots, e_0)$  eingefügt, da die Pfade  $(m_i, \dots, e_i)$  für  $i \in \{1, \dots, r\}$  nicht die minimale Höhe der Zeichnung sicherstellen, obwohl dies für den Beweis notwendig ist. Das Problem ergibt sich aus den Pfaden  $(m_i, \dots, e_i)$ , deren Kanten  $(m_i, x_i)$  um eins verkürzt werden könnten und somit die Höhe des Baumes verringert würde. Der Pfad  $(m_0, \dots, e_0)$  stellt die minimale Höhe von 49 in der Zeichnung von  $T(E)$  sicher. Des Weiteren wird die minimale Breite  $c^2 + (29 + n)c + 2$  durch den horizontalen vollen Pfad  $(m_0, \dots, e_r)$  bestimmt. In Abbildung 3.36(a) ist das der Pfad  $(m_0, \dots, e_2)$ . Die Argumentation des Beweises wird analog zum Beweis von Lemma 3.4 ausgeführt, wobei die Flächenschranke  $A = 49 \cdot (c^2 + (29 + n)c + 2)$  ist.  $\square$

**Lemma 3.5.** *Sei  $E$  ein boolescher Ausdruck in 3-KNF mit  $c$  Klauseln und  $n$  Variablen.  $E$  ist genau dann erfüllbar, wenn es eine ordnungserhaltende Zeichnung  $\Gamma(T(E))$  des geordneten Ternärbaumes  $T(E)$  auf dem 4-er Gitter mit höchstens der Fläche  $A = 40 \cdot W$  gibt, wobei  $W = 0.5r^2 + (20.5 + n)r + 2$ .*

*Beweis.* Im Folgenden wird analog zu den Beweisen in Lemma 3.3 und Lemma 3.4 dieselbe Beweistechnik für Ternärbäume auf dem 4-er Gitter angewendet. Ebenso, wie in Lemma 3.4 wird ein neuer globaler vertikaler voller Pfad  $(m_0, \dots, e_0)$  eingefügt. Da der vertikale neue volle Pfad  $(m_0, \dots, e_0)$  die minimale Höhe bestimmt, kann die horizontal nach rechts ausgehende Kante von  $m_0$  nicht kürzer als mit Länge zwei gezeichnet werden. Als Beispiel zeigt Abbildung 3.36(b) eine  $O_4$ -Zeichnung des Ternärbaumes  $T(E)$ .



(a)  $k=6$  ( $A = 49 \cdot W$  mit  $W = c^2 + (29 + n)c + 2$ )



(b)  $k=4$  ( $A = 40 \cdot W$  mit  $W = 0.5c^2 + (20.5 + n)c + 2$ )

Abbildung 3.36.  $O_k$ -Zeichnungen von  $T(E)$  für  $k \in \{4, 6\}$

Betrachte den obersten horizontalen vollen Pfad  $(m_0, \dots, m_c)$  in einer Zeichnung mit der minimalen Breite  $x$  und dem vertikalen Pfad  $(m_0, \dots, e_0)$  mit der Höhe  $y$ , die beide am Knoten  $m_0$  beginnen. Die minimale Breite der Zeichnung ist  $x$ , was aber zur Folge hat, dass die Höhe  $y+1$  ist. Zeichnet man den vertikalen Pfad  $(m_0, \dots, e_0)$  mit Höhe  $y$ , dann erzwingt man eine Breite des Pfades  $(m_0, \dots, m_c)$  von  $x+1$  (siehe Abbildung 3.36(b)). Wenn  $E$  mindestens zwei Klauseln enthält, dann ist die Breite größer als die Höhe, also  $x > y$ . Damit erhält man mit der letzteren Lösung die kleinere Fläche  $A = (x+1) \cdot y$ . Somit sind alle Kantenlängen dieser beiden vollen Pfade für mindestens zwei Klauseln bestimmt. Der restliche Beweis kann analog zum Beweis auf dem 6-er Gitter für die Fläche  $A = 40 \cdot (0.5r^2 + (20.5 + n)r + 2)$  durchgeführt werden.  $\square$

Somit ergibt sich der folgende Satz für ordnungserhaltende Zeichnungen auf  $k$ -Gittern für  $k \in \{4, 6, 8\}$ .

**Satz 3.13.** *Sei  $k \in \{4, 6, 8\}$ ,  $T$  ein gewurzelter  $(k-1)$ -näherer Baum und  $A \geq 1$ . Es ist  $\mathcal{NP}$ -hart zu entscheiden, ob eine  $O_k$ -Zeichnung  $\Gamma_k(T)$  mit Fläche kleiner oder gleich  $A$  existiert.*

**$OL_k$ -Zeichnungen** In diesem Abschnitt werden  $OL_k$ -Zeichnungen von Bäumen betrachtet. Sei  $T = (V, E)$  ein  $(k-1)$ -näherer Baum. In  $OL_k$ -Zeichnungen haben die ausgehenden Kanten eines Knotens  $v$  mit  $v \in V$  die gleiche Länge bezüglich der Maximummetrik. Es wird dargelegt, dass das Problem eine Zeichnung mit minimaler Fläche zu berechnen für den Fall  $k=4$   $\mathcal{NP}$ -hart bleibt. Für die Fälle  $k=6$  und  $k=8$  kann die bisher verwendete Beweistechnik nicht angewendet werden.

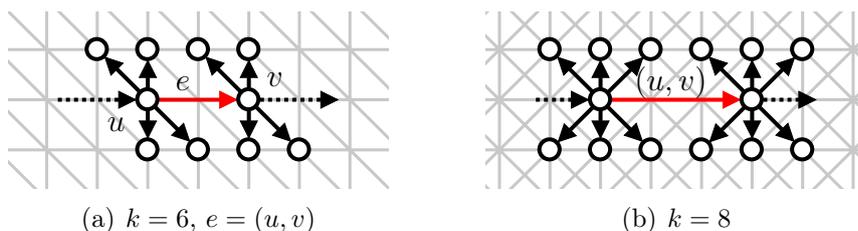
Zuerst werden Ternärbäume auf dem 4-er Gitter betrachtet. Die Konstruktion für  $OL_4$ -Zeichnungen von Ternärbäumen auf dem 4-Gitter ist wieder ähnlich zur Konstruktion von  $O_4$ -Zeichnungen. Der Unterschied dazu ist aber, dass die Eigenschaft der lokalen Uniformität erzwingt, dass Freiräume zwischen den vertikalen vollen Pfaden eingebaut werden müssen. Diese Tatsache führt zu anderen Abmessungen der Höhe und Breite und somit für die Fläche. Ansonsten kann der Beweis wie oben beschrieben durchgeführt werden.

**Satz 3.14.** *Sei  $T$  ein geordneter Ternärbaum und  $A \geq 1$ . Es ist  $\mathcal{NP}$ -hart zu bestimmen, ob eine  $OL_4$ -Zeichnung mit höchstens Fläche  $A$  existiert.*

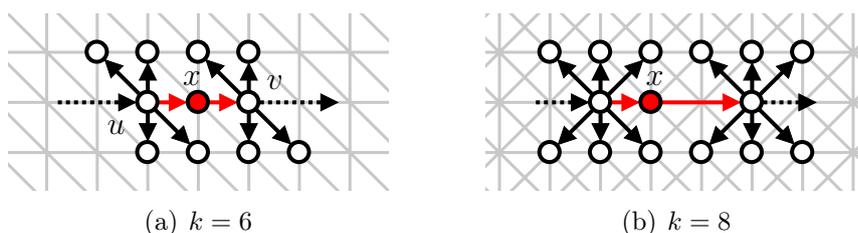
*Beweis.* In  $OL_4$ -Zeichnungen haben die ausgehenden Kanten eines Knotens dieselbe Länge. Die ausgehenden Kanten eines vollen Knotens  $v$  mit mindestens zwei vollen Knoten als Kinder können nicht kürzer als zwei sein. Deshalb kann der oberste horizontale Pfad nicht kürzer gezeichnet werden als in Abbildung 3.37 gezeigt. Es wird ein linker vertikaler voller Pfad konstruiert, der nicht niedriger gezeichnet werden kann, wie in Abbildung 3.37 dargestellt. Somit ist die minimale Breite und Höhe der Zeichnung bereits vorgegeben. Um die lokale Uniformität zu erfüllen und die Planarität zu erhalten werden Zwischenräume zwischen den Pfaden eingefügt. Der restliche Beweis wird wieder analog zu den Vorarbeiten und dem Beweis von Satz 3.13 mit der Fläche  $A = 62 \cdot (0.5c^2 + (25.5 + n)c + 3)$  durchgeführt.  $\square$



**Abbildung 3.37.**  $OL_4$ -Zeichnung des Ternärbaum  $T(E)$  auf dem 4-er Gitter für den booleschen Ausdruck  $E = (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_4})$  und Belegung  $\tau(x_1) = \tau(x_2) = false$  und  $\tau(x_3) = \tau(x_4) = true$



**Abbildung 3.38.** Zwei adjazente volle Knoten  $u, v$  die nicht lokal uniform gezeichnet werden können



**Abbildung 3.39.** Lokal uniforme Zeichnungen durch das Aufsplitten der Kante  $(u, v)$  aus Abbildung 3.38

Für  $OL_6$ - und  $OL_8$ -Zeichnungen kann der eingeführte Ansatz nicht so einfach transformiert werden wie für  $OL_4$ -Zeichnungen. Die lokale Uniformität lässt es nicht zu, mehrere volle Knoten so hintereinander zu platzieren und dadurch mit einem vollen Pfad die notwendigen Richtungen zu erzwingen. Wären zwei volle Knoten durch die Kante  $(u, v)$  verbunden, wobei die ausgehenden Kanten von  $u$  und  $v$  die Eigenschaft der lokalen Uniformität mit gleichen Längen erfüllen, so würden sich die Kinder von  $u$  und  $v$  überlagern, was die Planarität verletzt. In Abbildung 3.38(a) (Abbildung 3.38(b)) kann die ausgehende Kante von  $u$  nicht kürzer gezeichnet werden als mit der Länge zwei (drei). Würde man allerdings die ausgehenden Kanten von  $u$  mit der Länge  $x \in \mathbb{N}$  zeichnen und die ausgehenden Kanten von  $v$  mit Länge  $y \in \mathbb{N}$  mit  $x \neq y$ , dann lässt sich eine analoge Konstruktion nicht mehr durchführen.

Ein Ansatz für  $k = 6$  ( $k = 8$ ) wäre die Kante  $e = (u, v)$  zwischen zwei vollen Knoten  $u$  und  $v$  in zwei Teile aufzusplitten, so dass alle ausgehenden Kanten von  $u$  die Länge eins besitzen und die eingehende Kante in  $v$  Länge eins (zwei) besitzt, siehe Abbildung 3.39(a) (Abbildung 3.39(b)). Doch durch das Aufsplitten der Kante  $(u, v)$  in  $(u, x)$  und  $(x, v)$  können die Kantenrichtungen in der Zeichnung nicht mehr vorherbestimmt werden. Folglich ist der verwendete Beweisansatz für  $OL_k$ -Zeichnungen mit  $k \in \{6, 8\}$  nicht einfach übertragbar.

**$OP_k$ -Zeichnungen** Nachfolgend werden ordnungserhaltende Pattern- Zeichnungen betrachtet, d.h. die ausgehenden Kanten haben exakt die Richtung, die

durch die Pattern vorgegeben sind, siehe Definition 3.7. In diesem Fall kann für  $k \in \{4, 6, 8\}$  analog zu  $O_k$ -Zeichnungen bewiesen werden, dass es  $\mathcal{NP}$ -hart ist die Fläche einer minimalen Zeichnung eines  $(k - 1)$ -nären Baumes auf dem  $k$ -Gitter zu bestimmen.

**Satz 3.15.** *Sei  $k \in \{4, 6, 8\}$ ,  $T$  ein gewurzelter  $(k - 1)$ -närer Baum und  $A \geq 1$ . Das Bestimmen der Existenz einer  $OP_k$ -Zeichnung  $\Gamma_k(T)$  mit höchstens der Fläche  $A$  ist  $\mathcal{NP}$ -hart.*

*Beweis.* Die Konstruktion für  $OP_k$ -Zeichnungen mit  $k \in \{4, 6, 8\}$  ist identisch zu den Fällen für  $O_k$ -Zeichnungen, da diese Konstruktionen bereits für die wesentlichen Kanten  $OP_k$ -Zeichnungen erzwingen. Die unwesentlichen Kanten können problemlos nach den Pattern ausgerichtet werden, so dass der Beweis analog durchgeführt werden kann.  $\square$

**$OLP_k$ -Zeichnungen** Als eingeschränkste Zeichenart kann auch für  $OLP_k$ -Zeichnungen bewiesen werden, dass es  $\mathcal{NP}$ -hart ist eine Zeichnung mit minimaler Fläche für  $k \in \{4, 6, 8\}$  zu bestimmen. Diese Zeichnungen sind ordnungserhaltend, lokal-uniform und folgen den angegebenen Pattern.

**Satz 3.16.** *Sei  $k \in \{4, 6, 8\}$ ,  $T$  ein gewurzelter  $(k - 1)$ -närer Baum und  $A \geq 1$ . Das Bestimmen der Existenz einer  $OLP_k$ -Zeichnung  $\Gamma_k(T)$  mit höchstens der Fläche  $A$  ist  $\mathcal{NP}$ -hart.*

*Beweis.* Im Beweis von Satz 3.14 zu  $OL_4$ -Zeichnungen werden bereits  $OLP_4$ - Zeichnungen konstruiert, so dass die Aussage automatisch für  $k = 4$  und ordnungserhaltenden lokal uniformen Pattern-Zeichnungen gilt.

Um die Aussage für  $OLP_k$ - Zeichnungen mit  $k \in \{6, 8\}$  zu beweisen, wird die Reduktion des Beweises von Satz 3.13 angepasst. Jede Kante  $e = (u, v)$ , die zwei volle Knoten verbindet, wird durch einen neuen Knoten  $x$  in zwei Teile  $(u, x)$  und  $(x, v)$  aufgeteilt, siehe Abbildung 3.39. Aufgrund der lokalen Uniformität werden die ausgehenden Kanten von  $u$  mit Länge eins gezeichnet. Im Gegensatz zu den obigen Konstruktionen sind durch die Pattern die Richtungen der ausgehenden Kanten vollständig bestimmt. Für jeden neu eingeführten Knoten  $x$  hat die ausgehende Kante  $(x, v)$  dieselbe Richtung wie die eingehende  $(u, x)$ . Als Folge der Konstruktion haben die Zeichnungen der Bäume  $T(E)$  für  $k = 6$  ( $k = 8$ ) exakt dieselbe Abmessung wie die entsprechenden  $O_k$ -Zeichnungen. Der Rest des Beweises kann dazu analog durchgeführt werden.  $\square$

### 3.2.3 Ungeordnete Bäume

Im Gegensatz zu ordnungserhaltenden Zeichnungen (siehe Abschnitt 3.2.2) werden in diesem Abschnitt Zeichnungen von ungeordneten Bäumen auf  $k$ -Gittern betrachtet. In diesem Fall gibt es keine vorgegebene Reihenfolge der Kinder bzw. adjazenter Knoten in der Zeichnung. Diese Zeichnungen werden  $U_k$ -Zeichnungen genannt.

Analog zum geordneten Fall werden nicht-ordnungserhaltende lokal uniforme Zeichnungen als  $UL_k$ -Zeichnungen, Pattern-Zeichnungen als  $UP_k$ -Zeichnungen und lokal uniforme Pattern-Zeichnungen als  $ULP_k$ -Zeichnungen bezeichnet.

### 3.2.3.1 Fläche

Zuerst werden für verschiedene Zeichenarten Schranken für die benötigte Zeichenfläche gezeigt. Dabei können ebenfalls Unterschiede zum gewöhnlichen 4-er Gitter festgestellt werden.

**$U_k$ -Zeichnungen** Sei  $T$  ein ungeordneter Binärbaum mit  $n$  Knoten, der auf dem 4-er Gitter gezeichnet werden soll. Dann gibt es einen Linearzeitalgorithmus, der eine  $hv$ -Zeichnung in der Fläche  $\mathcal{O}(n \log n)$  erzeugt [CDBP92].

Ein *Fibonacci*baum  $T_{fib}^h$  mit Höhe  $h$  wird wie folgt rekursiv definiert: Der Fibonaccibaum  $T_{fib}^0$  mit Höhe 0 ist der Baum mit einem Knoten und  $T_{fib}^1$  mit Höhe 1 ist der Baum aus einem Knoten mit einem Kind. Der Fibonaccibaum der Höhe  $h \in \mathbb{N}$  besteht aus einer Wurzel mit zwei Kindern  $u$  und  $v$ , wobei  $u$  die Wurzel von  $T_{fib}^{h-1}$  und  $v$  die Wurzel von  $T_{fib}^{h-2}$  ist. Fibonaccibäume können mittels  $hv$ -Zeichnungen in Linearzeit in der Fläche  $\mathcal{O}(n)$  gezeichnet werden [CDBP92]. Im Allgemeinen benötigen Aufwärtszeichnungen von Binärbäumen eine Fläche in  $\Omega(n \log n)$  [CDBP92]. Die Erweiterung von  $hv$ -Zeichnungen auf dem 4-er Gitter sind  $hvd_6$ -Zeichnungen auf dem 6-er Gitter. Für vollständige Ternärbäume wurde in Satz 3.3 bereits eine obere und untere Schranke in  $\Theta(n^{1.262})$  Fläche gezeigt.

**Satz 3.17.** *Es existiert ein Linearzeitalgorithmus zum Zeichnen einer  $hvd_6$ -Zeichnung eines ungeordneten beliebigen Ternärbaumes auf dem 6-Gitter in der Fläche  $\mathcal{O}(n^{1.631})$ .*

*Beweis.* Die Breite einer Zeichnung  $\Gamma(T)$  eines Baumes  $T$  ist  $W(T)$ . Seien  $T_1, T_2$  und  $T_3$  Unterbäume von  $T$  mit der Wurzel  $r$  und mit  $W(T_1) \leq W(T_2) \leq W(T_3)$ . Die Zeichnung wird rekursiv erstellt mit der Wurzel in der linken oberen Ecke. Relativ zu  $r$  wird die Wurzel von  $T_1$  auf die Koordinaten  $(1, 1)$ ,  $T_2$  horizontal rechts mit dem Abstand  $2 + W(T_1)$  und  $T_3$  vertikal unterhalb eingefügt, siehe Abbildung 3.40.

Wenn die Breite einer  $hvd_6$ -Zeichnung minimiert wird und o. B. d. A. die Wurzel von  $T_1$  auf die Koordinaten  $(1, 1)$  platziert wird, dann ergibt sich für die Höhe höchstens  $\mathcal{O}(n)$ . Für die Breite von  $T$  ergibt sich  $W(T) = \max\{2 + W(T_1) + W(T_2), W(T_3)\}$ . Daraus ergibt sich  $W(T) = \mathcal{O}(n^{\log_3 2}) = \mathcal{O}(n^{0.631})$ , was durch die Berechnungen von Frati im Beweis von Theorem 5 in [Fra07] bestätigt wird.  $\square$

**Satz 3.18.** *Die obere und untere Schranke für die Fläche einer  $hvd_6$ -Zeichnung eines ungeordneten beliebigen Ternärbaumes mit  $n$  Knoten ist in  $\Theta(n^{1.631})$ .*

*Beweis.* Die obere Schranke folgt direkt aus Satz 3.17. Um die untere Schranke zu zeigen, wird ein Ternärbaum mit einem Pfad der Länge  $n/2$  betrachtet, an dem ein Unterbaum (vollständiger Ternärbaum) mit  $n/2$  Knoten hängt. Der Pfad benötigt

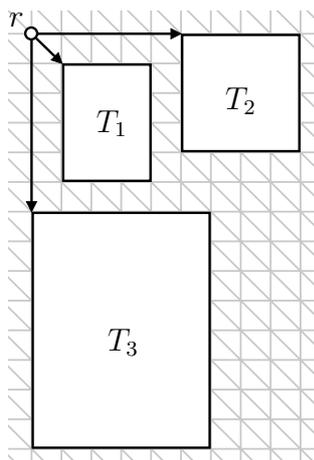


Abbildung 3.40. Skizze für Satz 3.17

in einer  $hvd_6$ -Zeichnung in  $\Omega(n)$  in mindestens eine Richtung und der vollständige Unterbaum benötigt  $\Omega(n^{0.631})$  in jede Richtung.  $\square$

Diese Ergebnisse sind vergleichbar mit den Ergebnissen für  $hv$ -Zeichnungen von Binärbäumen auf dem orthogonalen Gitter, wo die Schranke für vollständige  $\Theta(n)$  und für beliebige Binäräume  $\Theta(n \log n)$  ist.

Auf dem 4-er Gitter wurde ein Ansatz des rekursiven Windens („recursive winding“), mit dem sich Binärbäume zeichnen lassen, präsentiert. Eine Version des Verfahrens erstellt aufwärtsplanare Zeichnungen mit Rechteck Unterbaum Separation, der Höhe in  $\mathcal{O}(\frac{n}{A} \log A)$  und der Breite in  $\mathcal{O}(A \log n / \log A)$  mit  $A \in \{2, \dots, n\}$  [CGKT02, CGKT97].

Für das 8-er Gitter kann das Verfahren angepasst werden. Jedoch ist die von Chan et al. erreichte Fläche nur mit Kantenknicken zu erreichen. Denn aufgrund der diagonalen Kantenrichtung und der linearen Abhängigkeit der möglichen Kantenrichtungen ist es nicht möglich diese Fläche mit geradlinigen Kanten zu erreichen. Dies kann umgangen werden, indem höchstens linear viele Knicke in der Anzahl der Knoten eingeführt werden. In Abbildung 3.42(a) ist eine Zeichnung des Binärbaumes nach Anwendung des Verfahrens von Chan et al. auf dem 4-er Gitter gezeigt.

**Satz 3.19.** *Sei  $T$  ein Ternärbaum mit  $n$  Knoten. Es gibt eine planare Aufwärtszeichnung auf dem 8-er Gitter von  $T$  mit Höhe in  $\mathcal{O}(\frac{n}{A} \log A)$  und einer Breite in  $\mathcal{O}(\frac{A \log n}{\log A})$  mit  $A \in \{2, \dots, n\}$  und weniger als  $n - |\text{Blätter}(T)|$  Knicken.*

*Beweis.* Gegeben sei ein Ternärbaum  $T_3 = (V_3, E_3)$  mit  $n$  Knoten. Dieser Baum  $T_3$  wird in einen Binärbaum  $T_2 = (V_2, E_2)$  transformiert. Für jeden Knoten  $v \in V_3$  mit drei Kindern  $v_1, v_2, v_3$  werden die Kanten  $(v, v_1)$  und  $(v, v_2)$  ersetzt durch einen neuen Dummyknoten  $w$  mit den Kanten  $(v, w)$ ,  $(w, v_1)$  und  $(w, v_2)$ . Es ist zu beachten, dass  $|V_2|$  und  $|E_2|$  linear in der Anzahl der Knoten  $n$  sind.

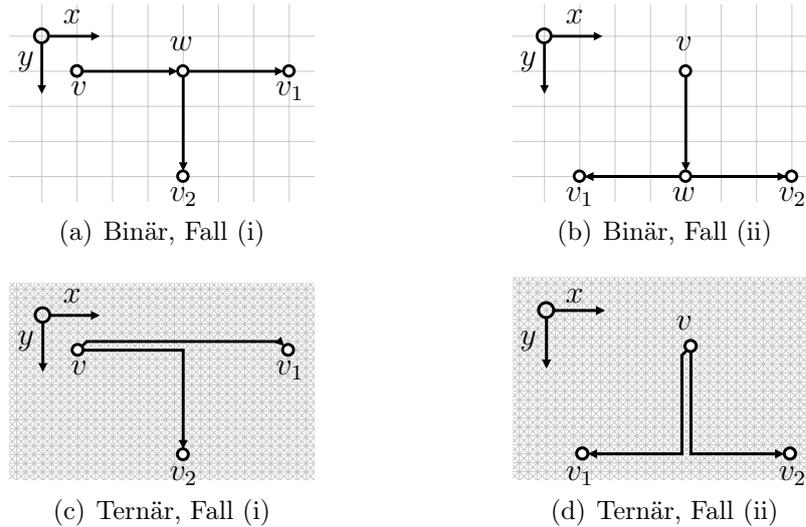


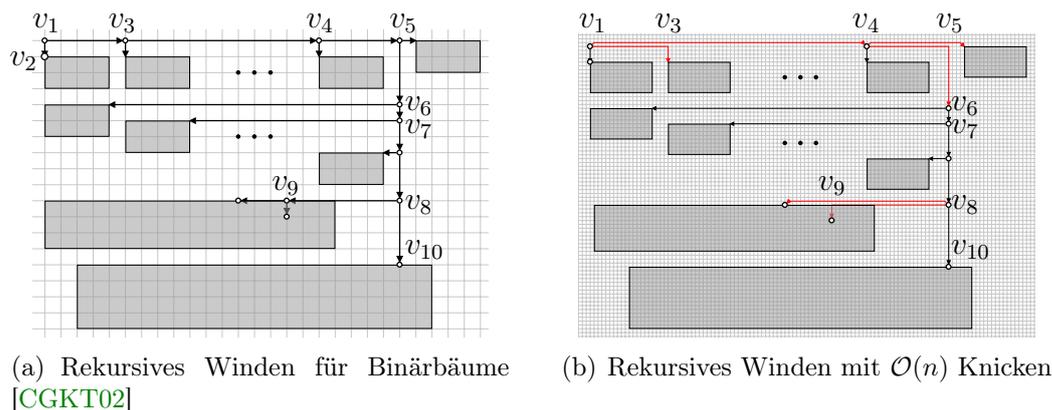
Abbildung 3.41. Ternärisierung

Im nächsten Schritt, wird der Binärbaum  $T_2$  auf dem 4-er Gitter mit dem Algorithmus von Chan et al. [CGKT97, CGKT02] gezeichnet. Daraus ergibt sich eine Zeichnung  $\Gamma(T_2)$  mit einer Höhe in  $\mathcal{O}(\frac{n}{A} \log A)$  und einer Breite in  $\mathcal{O}(\frac{A \log n}{\log A})$  mit  $A \in \{2, \dots, n\}$ . Jeder Knoten  $v \in V_2$  mit zwei Kindern hat die folgende Eigenschaft. Entweder, im Fall (i), hat eine ausgehende Kante von  $v$  die gleiche Richtung wie die eingehende Kante  $e$  von  $v$  und die andere ist orthogonal dazu, oder im Fall (ii) sind beide ausgehenden Kanten orthogonal zu  $e$  gezeichnet. Siehe dazu Abbildung 3.41(a) und 3.41(b).

Danach wird die geradlinige Zeichnung  $\Gamma(T_2)$  auf dem 4-er Gitter in eine Zeichnung  $\Gamma(T_3)$  auf dem 8-er Gitter mit Kantenknicken transformiert. Alle Koordinaten der nicht-Dummyknoten werden auf denselben Koordinaten wie in  $\Gamma(T_2)$  auf dem 8-er Gitter platziert. Jeder Dummyknoten  $w \in V_2$  und dessen inzidente Kanten  $(v, w)$ ,  $(w, v_1)$  und  $(w, v_2)$  werden ersetzt durch die entsprechenden Originalkanten  $(v, v_1)$  und  $(v, v_2)$  ersetzt. Dazu werden im Folgenden die Koordinaten der Knicke betrachtet. Alle restlichen geradlinig gezeichneten Kanten werden aus  $\Gamma(T_2)$  ohne Änderung übernommen.

Zuerst wird der Fall (i) betrachtet. Seien  $(x_u, y_u)$  die Koordinaten eines Knotens  $u$  in  $\Gamma(T_2)$ . O. B. d. A. zeigt eine Kante  $(v, w) \in E_2$  für einen Dummyknoten  $w$  in  $\Gamma(T_2)$  nach rechts,  $(w, v_1)$  hat ebenfalls diese Richtung und  $(w, v_2)$  zeigt nach unten, siehe Abbildung 3.41(a). Dann gilt  $y_v = y_w = y_{v_1}$  und  $x_w = x_{v_2}$ . In  $\Gamma(T_3)$  wird die Kante  $(v, v_1)$  mit zwei Knicken auf den Koordinaten  $(x_v + 0.25, y_v - 0.25)$  und  $(x_{v_1} - 0.25, y_{v_1} - 0.25)$  gezeichnet. Die Kante  $(v, v_2)$  wird mit einem Knick auf den Koordinaten  $(x_w, y_w)$  anstelle von  $w$  gezeichnet. Ein Beispiel ist in Abbildung 3.41(b) dargestellt.

Es bleibt der Fall (ii), wo o. B. d. A. die Kante  $(v, w)$  nach unten,  $(w, v_1)$  nach



**Abbildung 3.42.** Rekursives Winden von Binär- und Ternärbäumen

links und  $(w, v_2)$  nach rechts zeigt. In diesem Fall werden die Knicke von  $(v, v_1)$  auf die Koordinaten  $(x_v - 0.25, y_v + 0.25)$  und  $(x_w - 0.25, y_w)$  platziert. Für die Kante  $(v, v_2)$  wird der Knick an die Position des nicht mehr vorhandenen Dummyknotens  $w$  gesetzt, siehe Abbildungen 3.41(c) und 3.41(d).

Für alle restlichen Richtungen der eingehenden Kante eines Dummyknotens  $w$  ist die Konstruktion analog. Es ist zu beachten, dass die Konstruktion die Planarität und die Rechteck Unterbaum Separation erhält, die bereits durch die Zeichnung  $\Gamma(T_2)$  gegeben war.  $\Gamma(T_3)$  ist eine Zeichnung mit ganzzahligen Koordinaten für die Knoten und Knicken auf vielfachen von 0.25. Somit werden die Koordinaten der Knoten und Knicke vervierfacht, so dass eine Zeichnung auf ganzzahligen Koordinaten resultiert. Da die Fläche um das 16-fache wächst, erhöht sich die Fläche von  $\Gamma(T_3)$  im Vergleich zu  $\Gamma(T_2)$  nicht. Die Anzahl der eingeführten Kantenknicke ist insgesamt kleiner als  $2\frac{n}{3} + \frac{n}{3} = n$ , da ein Drittel der Kanten zwei Knicke und ein weiteres Drittel einen Knick erhält. Da jede inzidente Kante zu einem Blatt keinen Knick benötigt, ergeben sich  $n - |\text{Blätter}(T_3)|$  viele Kantenknicke. Eine schematische Zeichnung ist in Abbildung 3.42 dargestellt. □

Damit kann die obere Schranke mit Höhe in  $\mathcal{O}(\frac{n}{A} \log A)$  und einer Breite in  $\mathcal{O}(A \log n / \log A)$  mit  $A \in \{2, \dots, n\}$  für aufwärtsplanare Zeichnungen, die bereits für das von Chan et al. [CGKT02] entworfene Verfahren gelten, auch für nicht aufwärtsplanare Zeichnungen von Ternärbäumen auf dem 8-er Gitter mit  $\mathcal{O}(n)$  Knicken reproduziert werden. Für Zeichnungen von Ternärbäumen mit beliebigen Kantenrichtungen lässt sich das präsentierte Verfahren aus dem Beweis von Satz 3.19 anwenden.

**Korollar 3.8.** Sei  $T$  ein Ternärbaum mit  $n$  Knoten. Es gibt eine planare Zeichnung auf dem 8-er Gitter von  $T$  mit Höhe in  $\mathcal{O}(\frac{n}{A} \log A)$  und einer Breite in  $\mathcal{O}(A \log n + A)$  mit  $A \in \{2, \dots, n\}$  und weniger als  $n - |\text{Blätter}(T)|$  Knicken.

	$k = 4$	$k = 6$	$k = 8$
$U$	$\mathcal{NP}$ [BC87]	$\mathcal{NP}$ , Satz 3.22	$\mathcal{NP}$ , Satz 3.21
$UL$	$\mathcal{NP}$ , Satz 3.23	$\mathcal{NP}$ , Satz 3.23	$\mathcal{NP}$ , Satz 3.23
$UP$	$\mathcal{NP}$ , Satz 3.24	$(\mathcal{NP})$ , Verm. 3.2	$\mathcal{NP}$ , Satz 3.24
$ULP$	$\mathcal{NP}$ , Satz 3.24	$(\mathcal{NP})$ , Verm. 3.2	$\mathcal{NP}$ , Satz 3.24

**Tabelle 3.5.** Einheitliche Kantenlänge nicht ordnungserhaltender Zeichnungen  $\mathcal{NP}$ -hart

Wenn  $A = \lceil \log n \rceil$  gewählt wird, dann ergibt sich eine Zeichnung mit der Höhe in  $\mathcal{O}(\frac{n}{\log n} \log \log n)$ , Breite in  $\mathcal{O}(\log n)$  und liegt daher in einer Fläche in  $\mathcal{O}(n \log \log n)$ .

Nachdem das Zeichnen von Ternärbäumen auf dem 8-er Gitter betrachtet wurde, wird im folgenden Abschnitt die Komplexität von Zeichnungen mit einheitlichen Kantenlängen betrachtet.

### 3.2.3.2 Einheitliche Kantenlänge

In diesem Abschnitt werden Ergebnisse zur Komplexität von nicht-ordnungserhaltenden Zeichnungen von ungeordneten 7-nären Bäumen auf dem 8-er Gitter präsentiert. Nach Satz 3.8 gibt es immer eine planare Zeichnung eines vollständigen Baumes auf dem 8-er Gitter. Diese Konstruktion kann auch für Zeichnungen von beliebigen 7-nären Bäumen auf dem 8-er Gitter angewendet werden, indem in einem Top-Down Verfahren die ausgehenden Kanten eines Knotens  $u \in V$  mit der Länge  $3^{h(T)-t(u)-1}$  gezeichnet werden. Die gleiche Konstruktion kann für beliebige Bäume vom Grad acht verwendet werden. Somit gibt es für jeden solchen Baum auf dem 8-er Gitter eine planare Zeichnung.

Eine Übersicht der Komplexität, eine Zeichnung mit einheitlichen Kantenlängen zu bestimmen, liefert Tabelle 3.5. Für  $U_4$ -Zeichnungen wurde von Bhatt und Cosmadakis die  $\mathcal{NP}$ -Härte für Ternärbäume auf dem 4-er Gitter gezeigt [BC87]. Die restlichen Fälle auf  $k$ -Gittern werden im Folgenden genauer betrachtet.

**$U_k$ -Zeichnungen** Zuerst wird die Komplexität des Problems betrachtet, einen gegebenen Baum vom Grad acht oder einen 7-nären Baum auf dem 8-er Gitter mit einheitlicher Kantenlänge zu zeichnen. Im Folgenden wird nicht zwischen einem 7-nären Wurzelbaum und einem ungerichteten Baum vom Grad acht unterschieden, da diese bis auf die Kantenrichtungen isomorph sind. Zum Bestimmen der Komplexität des Zeichnens mit einheitlichen Kantenlängen wird eine Reduktion vom folgenden  $\mathcal{NP}$ -vollständigen Problem angegeben.

**Definition 3.11 (NOT-ALL-EQUAL-3-SAT, NAE3SAT).** [GJ90] Gegeben ist die Menge an Variablen  $X$ , die Menge an Klauseln  $F$ , bestehend aus drei dis-

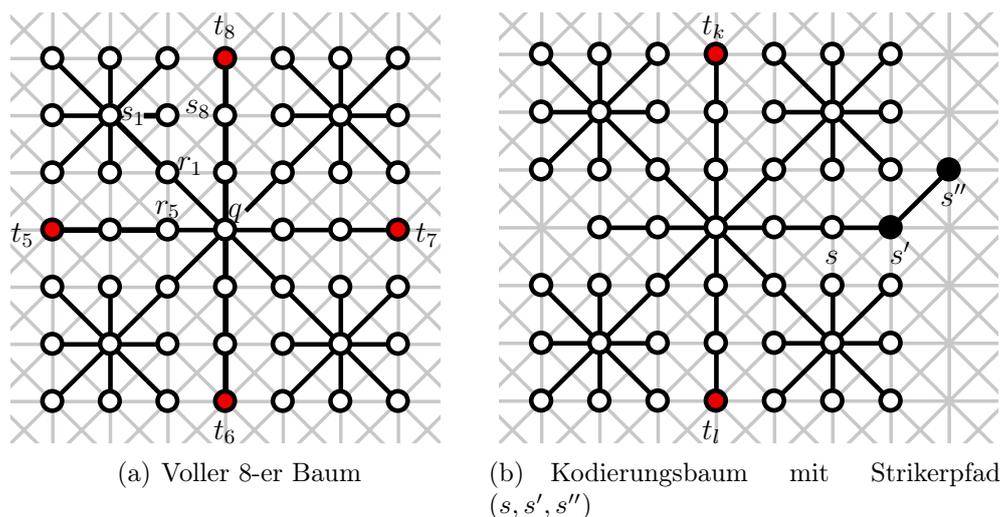


Abbildung 3.43. Teilbäume für die Reduktion

junktiv verknüpften Literalen und die konjunktive Verknüpfung der Klauseln in einem booleschen Ausdruck  $E$ , vgl. Definition 3.10. Gefragt ist, ob es eine Belegung  $\beta$  gibt, die dazu führt, dass in jeder Klausel mindestens ein Literal true und mindestens ein Literal false ist.

**Satz 3.20.** *NAE3SAT ist  $\mathcal{NP}$ -vollständig [GJ90].*

Es wird die Komplexität des Bestimmens von  $U_8$ -Zeichnungen mit einheitlicher Kantenlänge betrachtet. Dieses Problem ist offensichtlich genauso schwer, wie das Problem eine Zeichnung mit Kantenlänge eins zu bestimmen. Es wird von *NAE3SAT* reduziert und dabei in Polynomzeit aus einem gegebenen booleschen Ausdruck  $E$  in  $\beta$ -KNF ein Baum mit Grad acht konstruiert. Der boolesche Ausdruck  $E$  hat dabei  $n$  Variablen und  $c$  Klauseln.

Um die Beschreibung zu vereinfachen, wird ein *voller 8-er Baum* definiert. Dieser besteht aus einem Knoten  $q$  mit acht Nachbarn  $r_1, \dots, r_8$ , siehe Abbildung 3.43(a). Des Weiteren haben die Knoten  $r_1, \dots, r_8$  die entsprechenden adjazenten Knoten  $s_1, \dots, s_8$ . Die vier Knoten  $s_1, \dots, s_4$  haben sieben weitere adjazente Knoten, die *Eckblätter* genannt werden. Jeder, der übrigen Knoten  $s_5, \dots, s_8$  hat noch einen Nachbarn, genannt *Zentrumsblatt*. In Abbildung 3.43(a) sind die Blätter  $t_5, \dots, t_8$  die *Zentrumsblätter* und alle übrigen Blätter sind *Eckblätter*. Die *Position* eines solchen vollen 8-er Baumes wird mit der Position des Knotens  $q$  identifiziert.

**Lemma 3.6.** *Ein voller 8-er Baum hat exakt eine Zeichnung mit Kantenlänge eins auf dem 8-er Gitter abgesehen von Vertauschung und Knotenbenennung.*

*Beweis.* Wie oben erwähnt seien  $s_1, \dots, s_4$  die Nachbarn der Eckblätter und  $s_5, \dots, s_8$  die Nachbarn der *Zentrumsblätter* eines vollen 8-er Baumes. Angenommen, die Kan-

te  $(q, r_i)$  für  $i \in \{1, \dots, 4\}$  ist horizontal mit Länge eins gezeichnet. Notwendigerweise sind die restlichen sieben inzidenten Kanten von  $q$  ebenfalls mit Länge eins gezeichnet. Dann bleiben für  $(r_i, s_i)$  noch drei mögliche Richtungen, eine horizontal und zwei diagonal. Wenn  $(r_i, s_i)$  horizontal gezeichnet wird, dann können die sieben benachbarten Eckblätter nicht planar gezeichnet werden. Das ist auch der Fall, wenn  $(r_i, s_i)$  diagonal mit Kantenlänge eins gezeichnet wird. Daher kann  $(q, r_i)$  nicht horizontal und aus selbigem Grund auch nicht vertikal gezeichnet werden, sondern muss diagonal gezeichnet werden.

Für die Kante  $(r_i, s_i)$  bleiben fünf mögliche Richtungen. Angenommen, die Kante  $(r_i, s_i)$  hat eine andere Steigung wie  $(q, r_i)$ , dann überlagern sich die benachbarten Eckblätter von  $s_i$  mit den benachbarten Knoten von  $q$ . Deshalb muss das Quadrat, das der Knoten  $s_i$  und seine adjazenten Knoten aufgrund wegen Kantenlänge eins bilden, an der Ecke des  $7 \times 7$ -Quadrats gezeichnet werden. Andernfalls gäbe es eine Kante, die länger ist als eins. Letztendlich bleibt für die Pfade von  $q$  zu den Zentrumsblättern nur noch die Horizontale und die Vertikale übrig.  $\square$

Ein *8-er Kodierungsbaum* ist ein voller 8-er Baum, in dem zwei Zentrumsknoten weggelassen werden, siehe Abbildung 3.43(b) ohne  $s'$  und  $s''$ . Die beiden neuen Blätter werden *Strikerblätter* genannt. Ein *Striker* ist ein Pfad der Länge zwei, der an eines der beiden Strikerblätter angehängt wird, vgl. in Abbildung 3.43(b) Pfad  $(s, s', s'')$ . Betrachtet man eine Zeichnung mit Kantenlänge eins eines Kodierungsbaumes mit einer festen Position von  $s$ , dann ist die Position von  $s'$  festgelegt. Für das Blatt  $s''$  bleiben drei mögliche Gitterpunkte wegen Kantenlänge eins übrig. Diese befinden sich außerhalb des  $7 \times 7$ -Quadrats.

Im Folgenden werden die Bezeichnungen Knoten, Blatt und Pfad verwendet, um auch volle 8-er Bäume und 8-er Kodierungsbäume zu bezeichnen. Das Aneinanderfügen von vollen 8-er Bäumen und 8-er Kodierungsbäumen erfolgt, indem eine Kante zwischen die Zentrumsblätter eingefügt wird, eine so genannte *Zentrumsverbindung*. Oder es wird eine Kante zwischen Eckblättern eingefügt, die dann eine *Eckverbindung* herstellen. Es ist zu beachten, dass dies die einzigen Verbindungsmöglichkeiten sind, die im Folgenden zwischen diesen beiden Knotentypen verwendet wird.

Seien  $u$  und  $v$  zwei volle 8-er Bäume, die über eine *Zentrumsverbindung* adjazent sind. Aufgrund von Lemma 3.6 gibt es für die beiden vollen 8-er Bäume vier *relative Positionen*, nämlich  $u$  ist *links*, *rechts*, *oberhalb* oder *unterhalb* von  $v$  mit Kantenlänge eins. Sei die absolute Position von  $u$  im Gitter festgelegt und  $v$  hat eine relative Position zu  $u$ . Dann gibt es für  $v$  exakt drei mögliche absolute Positionen im Gitter unter Einhaltung von Kantenlänge eins. Als Beispiel sei  $u$  links von  $v$ , dann unterscheiden sich die  $y$ -Koordinaten von  $u$  und  $v$  höchstens um den Wert 1.

Für einen booleschen Ausdruck  $E$  in *3-KNF* mit  $c$  Klauseln und  $n$  Variablen wird ein Baum vom Grad acht  $S(n, c+1)$  konstruiert, siehe Abbildung 3.44. Zuerst wird ein *Elementarbaum*  $S(n, 1)$  definiert, der zum Einen aus einem Pfad von  $n+2$  vollen 8-er Bäumen  $(u_0, v_1, \dots, v_n, w_0)$  besteht. Dieser Pfad wird *Elementarpfad* genannt. Und zum Anderen werden über *Zentrumsverbindungen* Kodierungsbäume  $x_{i1}$  und  $\bar{x}_{i1}$  als Nachbarn an jeden vollen 8-er Baum  $v_i$  mit  $i \in \{1, \dots, n\}$  gefügt.

Vervollständigt wird ein Elementarbaum, indem drei zusätzliche volle 8-er Bäume  $u_1, u'_1, u''_1$  ( $w_1, w'_1, w''_1$ ) über Zentrumsverbindungen und zwei volle 8-er Bäume  $a_{11}$  und  $d_{11}$  ( $b_{11}$  und  $c_{11}$ ) über Eckverbindungen an den Knoten  $u_0$  ( $w_0$ ) gehängt werden. Die über Eckverbindungen angehängten vollen 8-er Bäume werden  $\alpha_{11}$  mit  $\alpha \in \{a, b, c, d\}$  bezeichnet.

Der Elementarbaum  $S(n, 1)$  bildet den Konstruktionsbeginn. Im Konstruktions-schritt  $j \rightarrow j + 1$  wird  $S(n, j)$  auf  $S(n, j + 1)$  erweitert, indem volle 8-er Bäume  $u_{j+1}, u'_{j+1}, u''_{j+1}, w_{j+1}, w'_{j+1}, w''_{j+1}$  an  $u_j, u'_j, u''_j, w_j, w'_j, w''_j$  mittels Zentrumsverbindungen angefügt werden. An die Kodierungsbäume  $x_{ij}$  und  $\bar{x}_{ij}$  werden mittels Zentrumsverbindungen ebenfalls Kodierungsbäume  $x_{i,j+1}$  und  $\bar{x}_{i,j+1}$  angefügt. Und für  $k \in \{1, \dots, j\}$  und  $\alpha \in \{a, b, c, d\}$  werden volle 8-er Bäume  $\alpha_{j+1,k}$  ( $\alpha_{k,j+1}$ ) an die Blätter  $\alpha_{jk}$  ( $\alpha_{kj}$ ) über Zentrumsverbindungen angefügt. Zuletzt wird ein voller 8-er Baum  $\alpha_{j+1,j+1}$  über eine Eckverbindung an  $\alpha_{jj}$  gehängt. Dieser Konstruktionsschritt wird so oft ausgeführt bis  $j + 1 = c$ .

Zum Schluss wird noch ein zusätzlicher, dem Konstruktionsschritt ähnlicher, Schritt ausgeführt. Anstatt Kodierungsbäume anzuhängen werden volle 8-er Bäume angefügt, um dann  $S(n, c + 1)$  zu erhalten. Dieser letzte Schritt stellt sicher, dass die freien Positionen jedes Kodierungsbaumes auf die gleiche  $y$ -Koordinate eingeschränkt sind. Im Baum  $S(n, c + 1)$  wird der Pfad  $(x_{i,c+1}, \dots, \bar{x}_{i,c+1})$  mit  $i \in \{1, \dots, n\}$  die  $i$ -te Spalte  $C_i$  genannt. Die Bäume, deren Knoten mit  $\alpha_{ij}$  mit  $\alpha \in \{a, b, c, d\}$  bezeichnet werden, werden im Folgenden  $\alpha$ -Begrenzungsbäume genannt.

**Lemma 3.7.** *Sei der Baum  $S(n, c + 1)$  auf dem 8-er Gitter mit Kantenlänge eins gezeichnet und sei  $(v_0 = u_0, v_1, \dots, v_n, w_0 = v_{n+1})$  der Elementarpfad. Dann haben alle  $v_i$  mit  $i \in \{0, \dots, n\}$  die gleiche relative Position zum jeweiligen Nachfolger  $v_{i+1}$ , entweder links von, rechts von, oberhalb oder unterhalb von  $v_i$ .*

*Beweis.* Betrachte den Elementarpfad  $(v_0 = u_0, v_1, \dots, v_n, w_0 = v_{n+1})$ . Dann hat der volle 8-er Baum  $u_0$  vier über Zentrumsverbindungen angefügte Nachbarn  $u_1, u'_1, u''_1$  und  $v_1$ . O. B. d. A. ist  $v_1$  rechts von  $u_0$  und  $u_1, u'_1, u''_1$  sind links von, oberhalb und unterhalb von  $u_0$  platziert, siehe Abbildung 3.44. Angenommen, der über eine Zentrumsverbindung zu  $v_1$  adjazente volle 8-er Baum  $v_2$  ist unterhalb (oberhalb) von  $v_1$  positioniert. Jeder der beiden vollen 8-er Bäume  $v_1$  und  $v_2$  hat jeweils vier über Zentrumsverbindungen angefügte Nachbarn. Dies führt zu einem Widerspruch, da so kein Platz für alle vier Nachbarn der beiden Knoten in der Zeichnung mit Kantenlänge eins vorhanden ist. Somit muss  $v_2$  rechts von  $v_1$  platziert werden. Analog kann dies für die restlichen vollen 8-er Bäume im Elementarpfad induktiv gezeigt werden.

Folglich müssen alle diese vollen 8-er Bäume des Elementarpfades nebeneinander gesetzt werden, wobei sich die  $y$ -Koordinaten zwischen benachbarten vollen 8-er Bäumen höchstens um 1 unterscheiden. Die über Zentrumsverbindungen angeschlossenen Nachbarn  $w_1, w'_1, w''_1$  von  $w_0$  sind o. B. d. A. symmetrisch zu den Nachbarn von  $u_0$  angeordnet, siehe Abbildung 3.44.  $\square$

Der Elementarpfad ist *horizontal eingebettet*, wenn alle aufeinanderfolgenden



Nachbarn  $v_i$  und  $v_{i+1}$  mit  $i \in \{0, \dots, n\}$  planar links (rechts) voneinander gezeichnet werden. Andernfalls ist er *vertikal eingebettet*. Sei  $\Gamma(S(n, c + 1))$  eine Zeichnung von  $S(n, c + 1)$  mit einem horizontal eingebetteten Elementarpfad. Dann unterteilt der Elementarpfad die Zeichnung  $\Gamma(S(n, c + 1))$  in eine *obere Hälfte* und eine *untere Hälfte*. Aufgrund der möglichen Vertauschung der inzidenten Kanten in der Zeichnung, ist der Pfad  $(x_{i1}, \dots, x_{i,c+1})$  entweder in der oberen Hälfte und  $(\bar{x}_{i1}, \dots, \bar{x}_{i,c+1})$  in der unteren Hälfte der Spalte  $C_i$  gezeichnet oder umgekehrt. Die Pfade  $(u_0, \dots, u_{c+1}), (u'_0, \dots, u'_{c+1}), (u''_0, \dots, u''_{c+1})$  werden *u-Pfade* von  $S(n, c + 1)$  genannt. Analog werden *w-Pfade* definiert.

**Lemma 3.8.** *Sei  $S(n, c + 1)$  mit Kantenlänge eins gezeichnet und sei der Elementarpfad horizontal auf dem 8-er Gitter eingebettet. Dann haben alle restlichen Kanten festgelegte Richtungen.*

*Beweis.* Sei  $\Gamma(S(n, 1))$  eine Zeichnung eines Elementarbaumes  $S(n, 1)$ , wobei der Elementarpfad  $(u_0, v_1, \dots, v_n, w_0)$  horizontal gezeichnet ist und  $v_1$  rechts von  $u_0$  platziert ist. Seien  $u_1$  links von  $u_0$ ,  $u'_1$  oberhalb von  $u_0$  und  $u''_1$  unterhalb von  $u_0$  platziert und seien die Nachbarn von  $w_0$  symmetrisch angebracht. Jeder volle 8-er Baum  $v_i$  mit  $i \in \{1, \dots, n\}$  hat zwei über Zentrumsverbindungen angeschlossene Kodierungsbäume  $x_{i1}$  und  $\bar{x}_{i1}$  als Nachbarn, die entweder oberhalb und unterhalb (oder umgekehrt) eingebettet sind. Alle über Zentrumsverbindungen angefügten vollen 8-er Bäume und Kodierungsbäume haben relative Positionen zueinander. Wegen Kantenlänge eins, kann der horizontale Abstand zwischen den über Eckverbindungen angebotenen vollen 8-er Bäumen  $\alpha_{11}$  mit  $\alpha \in \{a, b, c, d\}$  oberhalb (unterhalb) des Elementarpfades höchstens  $7(n + 2) + 1$  sein. Dies entspricht höchstens  $7(n + 2)$  freien zu besetzenden Gitterpunkten zwischen  $\alpha_{11}$  mit  $\alpha \in \{a, b, c, d\}$  oberhalb (unterhalb) des Elementarpfades. Da jeder volle 8-er Baum die Breite sieben Gitterpunkte abdeckt, belegt die horizontale Zeile der  $n$  Kodierungsbäume in der Summe mindestens  $7(n + 2)$  Gitterpunkte. Somit sind die horizontalen Positionen dieser Kodierungsbäume und der über Eckverbindungen angeschlossenen vollen 8-er Bäume festgelegt. Daher haben alle Kanten des Elementarbaumes  $S(n, 1)$ , bis auf die verbindenden Kanten zwischen den vollen 8-er Bäumen des Elementarpfades, eine festgelegte Steigung. Die gleiche Argumentation wird im Konstruktionsschritt  $j \rightarrow j + 1$  angewendet.  $\square$

O. B. d. A. ist der Elementarpfad immer horizontal von links nach rechts eingebettet, wie im Beweis von Lemma 3.8. Dann sind in jeder Spalte  $C_i$  mit  $i \in \{1, \dots, n\}$  die verbindenden Kanten der Pfade  $(x_{i,c+1}, \dots, \bar{x}_{i,c+1})$  vertikal gezeichnet. Jedes  $7 \times 7$ -Quadrat eines Kodierungsbaumes belegt exakt zwei Gitterpunkte mehr als der Kodierungsbaum selbst, siehe Abbildung 3.43(a). Diese freien Gitterpunkte werden verwendet, um die restliche Information des booleschen Ausdrucks  $E$  in den Baum  $S(n, c + 1)$  zu kodieren.

Betrachte Variable  $x_i$  und die  $j$ -te Klausel mit  $i \in \{1, \dots, n\}$  und  $j \in \{1, \dots, c\}$ . Wenn  $x_i$  nicht in Klausel  $j$  vorkommt, dann wird Striker  $(s_{ij}, s'_{ij}, s''_{ij})$  an den Kodierungsbaum  $x_{ij}$  und Striker  $(\bar{s}_{ij}, \bar{s}'_{ij}, \bar{s}''_{ij})$  an  $\bar{x}_{ij}$  gehängt. Wenn  $x_i$  nicht negiert

in Klausel  $j$  vorkommt, wird Striker  $(s_{ij}, s'_{ij}, s''_{ij})$  an  $x_{ij}$  gehängt. Und als letzter Fall, wenn die Variable negiert in  $x_i$  vorkommt, so wird Striker  $(\bar{s}_{ij}, \bar{s}'_{ij}, \bar{s}''_{ij})$  an  $\bar{x}_{ij}$  gehängt.

Im Folgenden wird die erläuterte Erweiterung des Baumes  $S(n, c + 1)$  mit  $T(E)$  bezeichnet. Es ist zu beachten, dass nach Lemma 3.8 in einer Zeichnung  $\Gamma(T(E))$  mit Kantenlänge eins alle Kanten bis auf die verbindenden Kanten des Elementarpfades und der zweiten Strikerkante  $(s'_{ij}, s''_{ij})$  oder  $(\bar{s}'_{ij}, \bar{s}''_{ij})$  festgelegte Steigungen besitzen.

In der Zeichnung  $\Gamma(T(E))$  wird der Kodierungsbaum  $z_{ij} \in \{x_{ij}, \bar{x}_{ij}\}$  und der entsprechende Striker  $S = (s, s', s'')$  betrachtet. Aufgrund der nicht notwendigen Erhaltung der Reihenfolge der Nachbarknoten kann  $z_{ij}$  vertikal gespiegelt werden, so dass  $S$  entweder auf die linke Seite gezeichnet wird oder auf die Rechte. Nach Lemma 3.7 unterscheiden sich die  $y$ -Koordinaten der Spalten  $C_i$  und  $C_{i-1}$  oder  $C_{i+1}$  mit  $i \in \{2, \dots, n-1\}$  um maximal 1. Wenn es einen freien Gitterpunkt auf der rechten Seite von  $z_{i-1,j}$  oder entsprechend auf der linken Seite von  $z_{i+1,j}$  gibt, dann kann der Knoten  $s''$  auf diesem Gitterpunkt unter Berücksichtigung der Kantenlänge eins eingebettet werden. Als Beispiel siehe den Kodierungsbaum  $\bar{x}_{22}$  in Abbildung 3.44 mit dem Striker  $S$ , der nach rechts gezeichnet ist. Es ist zu beachten, dass die Striker der Spalte  $C_1$  ( $C_n$ ) nur nach rechts (links) planar gezeichnet werden können.

**Lemma 3.9.** *Sei  $E$  ein boolescher Ausdruck in 3-KNF mit  $c$  Klauseln und  $n$  Variablen.  $E$  ist erfüllbar mit mindestens einem erfüllten und einem nicht-erfüllten Literal pro Klausel genau dann, wenn es eine Zeichnung  $\Gamma(T(E))$  mit Kantenlänge eins auf dem 8-er Gitter gibt.*

*Beweis.* “ $\Rightarrow$ ”: Sei  $\tau(E)$  eine erfüllende Belegung für den booleschen Ausdruck  $E$  mit  $n$  Variablen und  $c$  Klauseln. Der Baum  $T(E)$  wird, wie oben beschrieben, konstruiert. Um eine planare Zeichnung  $\Gamma(T(E))$  mit Kantenlänge eins zu erhalten, muss bestimmt werden, ob ein Pfad  $(x_{i1}, \dots, x_{i,c+1})$  in der oberen Hälfte und der entsprechende Pfad  $(\bar{x}_{i1}, \dots, \bar{x}_{i,c+1})$  in der unteren Hälfte eingebettet wird, oder umgekehrt. Wenn die Variable  $x_i$  mit  $i \in \{1, \dots, n\}$  mit *true* belegt ist, dann wird der entsprechende Pfad  $(x_{i1}, \dots, x_{i,c+1})$  in der oberen Hälfte eingebettet, andernfalls in der Unteren.

In der Zeile  $j$  oben (unten) für die Klausel  $j$  von  $\Gamma(T(E))$  gibt es ohne Striker  $n - 1$  freie Plätze, um entsprechend Striker planar einzubetten. Für die  $n - 3$  nicht-vorhandenen Variablen in Klausel  $j$  werden nach der Konstruktion in der  $j$ -ten Zeile oben (unten)  $n - 3$  Striker eingefügt. Da die Belegung  $\tau(E)$  sicherstellt, dass jede Klausel mindestens ein erfülltes und ein unerfülltes Literal besitzt, so müssen in jeder Zeile  $j$  oben (unten) mindestens ein Striker oder höchstens zwei Striker für vorhandene Variablen eingebettet sein. Das passt exakt zur Tatsache, dass in der  $j$ -ten Zeile in der oberen (unteren) Hälfte der Zeichnung für die  $j$ -te Klausel höchstens zwei Striker der entsprechenden Variablen der  $j$ -ten Klausel planar eingebettet werden können. Somit ist durch die gegebene erfüllte Belegung  $\tau(E)$  eine planare Zeichnung  $\Gamma(T(E))$  von  $T(E)$  mit Kantenlänge eins konstruiert worden.

“ $\Leftarrow$ ”: Sei  $\Gamma(T(E))$  eine Zeichnung mit Kantenlänge eins von  $T(E)$ . Nach Lemma 3.8 sind die Steigungen aller Kanten festgelegt bis auf die verbindenden Kanten

des Elementarpfades und der Strikerkanten. Ohne Striker gibt es, wie in “ $\Rightarrow$ ” bereits erwähnt,  $n - 1$  Löcher, d. h. zwei adjazente Gitterpunkte in der  $j$ -ten Zeile mit  $j \in \{1, \dots, c\}$  zwischen nebeneinander platzierten Kodierungsbäumen in der oberen Hälfte und ebenso  $n - 1$  Löcher in der unteren Hälfte. In jeder Zeile  $j$  (obere und untere Hälfte) sind  $n - 3$  Striker für nicht vorhandene Variablen der Klausel  $j$  angehängt. Deshalb können höchstens zwei Striker, die zu einer vorhandenen Variable in der aktuellen Klausel gehören, in der  $j$ -ten Zeile der oberen (unteren) Hälfte platziert werden. Für negierte und nicht-negierte Literale einer Klausel werden insgesamt drei Striker in den Baum eingefügt. Es ist nicht möglich, dass alle drei Striker in der oberen (unteren) Hälfte platziert werden. In einer planaren Zeichnung mit Kantenlänge eins müssen zwei Striker in der oberen und einer in der unteren Hälfte eingebettet sein, oder umgekehrt.

Ein Literal  $y_k$  mit  $k \in \{1, \dots, n\}$  ist entweder die Variable  $x_k$  oder seine Negation  $\bar{x}_k$ . Sei das Literal  $y_k$  in Klausel  $j$  vorhanden. Wenn  $y_k$  nicht negiert ist, dann wird das Literal auf erfüllt gesetzt, wenn der entsprechende Striker  $(s_{kj}, s'_{kj}, s''_{kj})$  in der oberen Hälfte der Zeichnung  $\Gamma(T(E))$  eingebettet ist. Andernfalls, wenn  $y_k$  negiert ist, dann ist das Literal nicht erfüllt, wenn der Striker  $(s_{kj}, s'_{kj}, s''_{kj})$  in der oberen Hälfte eingebettet ist. Somit erhält man eine erfüllende Belegung  $\tau'(E)$  hinsichtlich *NAE3SAT* mit mindestens einem erfüllten Literal und mindestens einem nicht erfüllten Literal in jeder Klausel.  $\square$

**Satz 3.21.** *Sei  $T$  ein Baum vom Grad acht. Es ist  $\mathcal{NP}$ -hart zu entscheiden, ob eine  $U_8$ -Zeichnung von  $T$  mit Kantenlänge eins existiert.*

Um die  $\mathcal{NP}$ -Härte für Zeichnungen mit einheitlicher Kantenlänge auf dem 6-er Gitter zu zeigen, wird im Folgenden die Idee vom 8-er und 4-er Gitter übernommen. Es sind entscheidende Änderungen notwendig, da zwei diagonale Richtungen des 8-er Gitters auf dem 6-er Gitter nicht verwendet werden können. Analog zum vollen 8-er Baum auf dem 8-er Gitter wird für das 6-er Gitter der *volle 6-er Baum* vom Grad sechs definiert, siehe Abbildung 3.45(a). Somit werden im Folgenden die nicht erlaubten diagonalen Kanten aus der Konstruktion eliminiert und eine Alternative für das 6-er Gitter beschrieben.

Der Knoten  $q$  besitzt Kanten zu den sechs adjazenten Knoten  $r_1, \dots, r_6$ . Jeder Knoten  $r_1, \dots, r_6$  hat weitere Nachbarn. Die Knoten  $r_2$  und  $r_5$  haben je drei adjazente Blätter und die Knoten  $r_3$  und  $r_6$  haben je ein adjazentes Blatt. Der Knoten  $r_1$  ( $r_4$ ) hat zwei benachbarte Knoten. Ein Knoten davon ist das Blatt  $s_1$  ( $s_4$ ) und der andere  $t_1$  ( $t_4$ ) mit zwei adjazenten Blättern. Die adjazenten Blätter von  $r_2$  und  $r_5$  werden analog zum Beweis auf dem 8-er Gitter wieder *Eckblätter* genannt. Als *Zentrumsblätter* werden die Knoten  $s_1, s_3, s_4$  und  $s_6$  bezeichnet, siehe Abbildung 3.45(a).

**Lemma 3.10.** *Ein voller 6-er Baum belegt in einer Zeichnung mit Kantenlänge eins auf dem 6-er Gitter höchstens ein  $5 \times 5$ -Quadrat von Koordinaten, wenn sich die Zentrumsblätter am Rand dieses  $5 \times 5$ -Quadrats befinden.*

*Beweis.* Sei der Knoten  $q$  eines vollen 6-er Baumes im Zentrum eines  $5 \times 5$ -Quadrats platziert, dann sind die Kanten  $(q, r_1), \dots, (q, r_6)$  mit Kantenlänge eins um  $q$  plat-

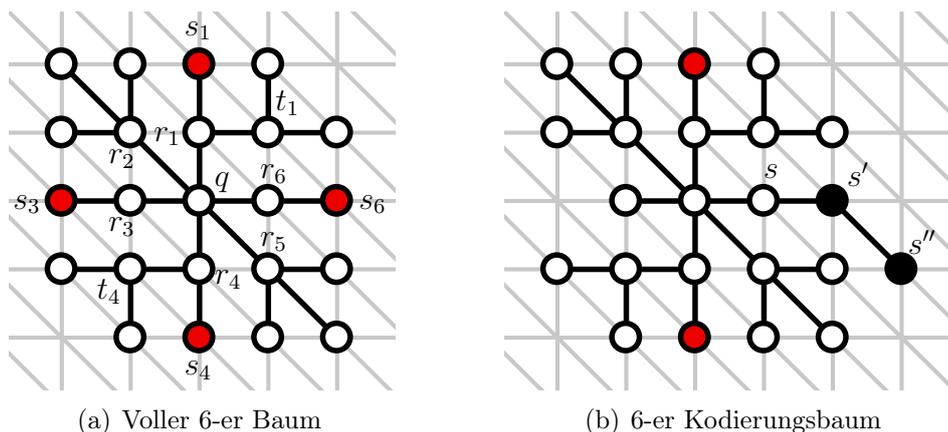


Abbildung 3.45. Voller 6-er Baum und 6-er Kodierungsbaum

ziert. Angenommen,  $(q, r_2)$  oder  $(q, r_5)$  ist horizontal gezeichnet, dann belegt eines der benachbarten Blätter die Diagonale, so dass nicht alle adjazenten Knoten von  $q$  in das 6-er Gitter planar eingebettet werden können. Folglich sind die beiden Kanten  $(q, r_2)$  und  $(q, r_5)$  diagonal eingezeichnet. Wenn sich die Zentrumsblätter  $s_1$  und  $s_4$  am oberen, linken, unteren oder rechten Rand des  $5 \times 5$ -Quadrats befinden, dann können die Knoten  $t_1, t_4$  und deren adjazente Blätter mit Kantenlänge eins nur innerhalb des  $5 \times 5$ -Quadrats platziert sein. Die benachbarten Blätter von  $r_3$  und  $r_6$  sind ebenfalls innerhalb des  $5 \times 5$ -Quadrates eingebettet.  $\square$

Ein *6-er Kodierungsbaum* ist ein voller 6-er Baum, in dem die Blätter  $s_3$  und  $s_6$  weggelassen werden, siehe Abbildung 3.45(b) ohne  $s'$  und  $s''$ . Analog zum 8-er Gitter wird der Pfad  $(s, s', s'')$  der Länge zwei *Striker* genannt. Der Striker wird an das *Strikerblatt*  $r_6$  angefügt.

Betrachtet man eine Zeichnung mit Kantenlänge eins eines 6-er Kodierungsbaumes mit den beiden Zentrumsblättern am Rand des  $5 \times 5$ -Quadrats und einer festen Position von  $s$ , dann ist die Position von  $s'$  festgelegt. Für das Blatt  $s''$  bleiben zwei mögliche Gitterpunkte übrig, die sich außerhalb des  $5 \times 5$ -Quadrats befinden und dabei die Eigenschaft der Kantenlänge eins erfüllen.

Im Folgenden werden die Bezeichnungen Knoten, Blatt und Pfad verwendet, um volle 6-er Bäume und 6-er Kodierungsbäume zu bezeichnen. Das Aneinanderfügen von vollen 6-er Bäumen und 6-er Kodierungsbäumen erfolgt, indem eine Kante zwischen die Zentrumsblätter eingefügt wird, eine so genannte *Zentrumsverbindung*. Dadurch wird sichergestellt, dass die Zentrumsblätter am Rand des  $5 \times 5$ -Quadrats gezeichnet sind, da sonst die Eigenschaft der Kantenlänge eins verletzt werden müsste. Oder es wird eine Kante zwischen Eckblättern eingefügt, die eine *Eckverbindung* herstellt. Es ist zu beachten, dass dies die einzigen Verbindungsmöglichkeiten sind, die für diese beiden Knotentypen (voller 6-er Baum, 6-er Kodierungsbaum) verwendet werden.

Wie auch auf dem 8-er Gitter sind zwei über eine Zentrumsverbindung adjazente

volle 6-er Bäume  $u$  und  $v$  mit Kantenlänge eins relativ zueinander positioniert. Entweder  $u$  liegt links, rechts, oberhalb oder unterhalb von  $v$ . Angenommen, der Knoten  $u$  hat eine feste Position, dann ergeben sich für  $v$  bei einer bestehenden relativen Position zwei Möglichkeiten.

Ähnlich zur Konstruktion des Baumes  $S(n, c + 1)$  auf dem 8-er Gitter wird die Konstruktion für das 6-er Gitter durchgeführt. Der *Elementarpfad*  $(u_0, v_1, \dots, v_n, w_0)$  wird analog zur Konstruktion auf dem 8-er Gitter definiert und besteht aus vollen 6-er Knoten, die über Zentrumsverbindungen zusammenhängen. Daraus kann der *Elementarbaum*  $S(n, 1)$  konstruiert werden. An jeden Knoten  $v_i$  mit  $i \in \{1, \dots, n\}$  werden die Kodierungsbäume  $x_{i1}$  und  $\bar{x}_{i1}$  über Zentrumsverbindungen angehängt. An  $u_0$  und  $w_0$  wird je ein voller 6-er Baum  $a_{11}$  und  $c_{11}$  über eine Eckverbindung angebunden. Es ist zu beachten, dass im Gegensatz zum 8-er Gitter nur die vollen 6-er Bäume  $a_{11}$  und  $c_{11}$  in  $S(n, 1)$  eingefügt werden. Darüberhinaus werden an  $u_0$  ( $w_0$ ) noch über Zentrumsverbindungen die Knoten  $u_1, u'_1$  und  $u''_1$  ( $w_1, w'_1$  und  $w''_1$ ) angefügt.

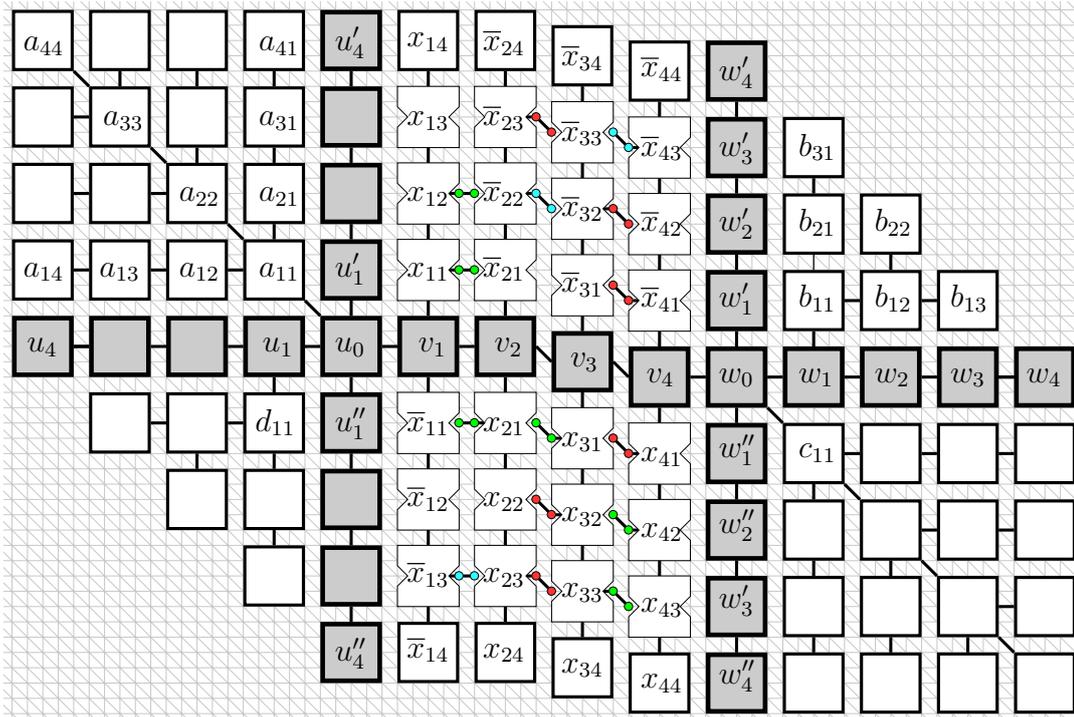
Der Baum  $S(n, 2)$  wird folgendermaßen definiert. An die Knoten  $u_1, u'_1$  und  $u''_1$  von  $S(n, 1)$  wird entsprechend je ein Knoten  $u_2, u'_2$  und  $u''_2$  angefügt. An die Kodierungsbäume  $x_{i1}$  und  $\bar{x}_{i1}$  werden über Zentrumsverbindungen Kodierungsbäume  $x_{i2}$  und  $\bar{x}_{i2}$  angefügt. Über Eckverbindungen werden an  $a_{11}$  ( $c_{11}$ ) über zwei Zentrumsverbindungen die Knoten  $a_{12}$  und  $a_{21}$  ( $c_{12}$  und  $c_{21}$ ) angebunden, sowie über eine Eckverbindung der Knoten  $a_{22}$  ( $c_{22}$ ). Zuletzt wird an den Knoten  $u_1$  der Knoten  $d_{11}$  und an  $w_1$  der Knoten  $b_{11}$  über eine Zentrumsverbindung angefügt.

Der Konstruktionsschritt  $j \rightarrow j + 1$  erfolgt ähnlich zum Konstruktionsschritt auf dem 8-er Gitter. Jedem Kodierungsbaum  $x_{ij}$  ( $\bar{x}_{ij}$ ) wird der Kodierungsbaum  $x_{i,j+1}$  ( $\bar{x}_{i,j+1}$ ) angefügt. Die Knoten  $u_i, u'_i$  und  $u''_i$  ( $w_i, w'_i$  und  $w''_i$ ) werden mit einer Zentrumsverbindung um  $u_{i+1}, u'_{i+1}$  und  $u''_{i+1}$  ( $w_{i+1}, w'_{i+1}$  und  $w''_{i+1}$ ) erweitert. Für  $k \in \{1, \dots, j\}$  und  $\alpha \in \{a, c\}$  werden volle 6-er Bäume  $\alpha_{j+1,k}$  ( $\alpha_{k,j+1}$ ) an die Blätter  $\alpha_{jk}$  ( $\alpha_{kj}$ ) über Zentrumsverbindungen angefügt. Zuletzt wird ein voller 8-er Baum  $\alpha_{j+1,j+1}$  über eine Eckverbindung an  $\alpha_{jj}$  gehängt. Jedem Knoten  $\beta_{j-\gamma,\gamma}$  mit  $\beta \in \{b, d\}$  und  $\gamma \in \{1, \dots, j\}$  wird über eine Zentrumsverbindung  $\beta_{j-\gamma+1,\gamma}$  angefügt und zusätzlich der neue Knoten  $\beta_{1,j}$  wird mit dem Knoten  $\beta_{1,j-1}$  verbunden, siehe dazu Abbildung 3.46. Die verbleibende Konstruktion von  $S(n, j + 1)$  aus  $S(n, j)$  verläuft analog zur Konstruktion auf dem 8-er Gitter. Der Konstruktionsschritt wird so oft ausgeführt bis  $j + 1 = c$ .

Zum Schluss wird noch ein zusätzlicher Konstruktionsschritt ausgeführt. Anstatt 6-er Kodierungsbäume zu verwenden, werden volle 6-er Bäume an die Kodierungsbäume angefügt, um dann  $S(n, c + 1)$  zu erhalten. Wie auch auf dem 8-er Gitter, stellt dieser letzte Schritt sicher, dass die freien Positionen jedes Kodierungsbaumes auf die gleichen  $y$ -Koordinaten eingeschränkt sind.

Im Baum  $S(n, c + 1)$  wird der Pfad  $(x_{i,c+1}, \dots, \bar{x}_{i,c+1})$  mit  $i \in \{1, \dots, n\}$  die  $i$ -te Spalte  $C_i$  genannt. Die Bäume für  $\alpha \in \{a, c\}$ , deren Knoten mit  $\alpha_{ij}$  benannt sind, werden im Folgenden  $\alpha$ -Begrenzungsbäume genannt. Analog für  $\beta \in \{b, d\}$  werden  $\beta$ -Begrenzungsbäume definiert.

Die Aussage von Lemma 3.7 kann mit der soeben beschriebenen Konstruktion



**Abbildung 3.46.**  $T(E)$  von  $E = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$  ( $n = 4$  und  $r = 3$ ) und Belegung  $x_1 = true, x_2 = x_3 = x_4 = false$  auf dem 6-er Gitter

analog für das 6-er Gitter übernommen werden.

**Lemma 3.11.** *Sei  $S(n, c + 1)$  mit Kantenlänge eins gezeichnet und sei der Elementarpfad horizontal auf dem 6-er Gitter eingebettet. Dann haben alle restlichen Kanten festgelegte Richtungen.*

*Beweis.* Wie im Beweis von Lemma 3.7 sei  $\Gamma(S(n, 1))$  eine Zeichnung eines Elementarbaumes  $S(n, 1)$  mit dem horizontal eingebetteten Elementarpfad  $(u_0, \dots, w_0)$  und sei  $v_1$  rechts von  $u_0$  angebracht. Seien o. B. d. A.  $u_1$  links von  $u_0$ ,  $u'_1$  oberhalb von  $u_0$  und  $u''_1$  unterhalb von  $u_0$  platziert. Analog sei dies für die adjazenten Knoten von  $w_0$  ausgeführt. Aufgrund der Kantenlänge eins und der nur sechs zur Verfügung stehenden Kantenrichtungen in der Zeichnung des Elementarbaumes  $\Gamma(S(n, 1))$ , können die Kodierungsbäume oberhalb des Elementarpfades in  $\Gamma(S(n, 1))$  nur zwischen den vollen Bäumen  $u_1$  und  $w_1$  gezeichnet sein, da die Kanten  $(u_0, u_1)$  und  $(w_0, w_1)$  nur vertikal nach oben verlaufen können. Aufgrund der fehlenden diagonalen Richtung nach rechts oben kann die Kante  $(w_0, w_1)$  nur vertikal nach oben verlaufen, wenn die horizontale Richtung nach rechts durch  $(w_0, w'_1)$  und die vertikale Richtung nach unten durch  $(w_0, w''_1)$  belegt ist. Somit bleiben den Kodierungsbäumen oberhalb von  $S(n, 1)$  nur  $5 \cdot n$  Gitterpunkte zwischen  $u_1$  und  $w_1$ . Da die  $n$  Kodierungsbäume in der Horizontalen fünf Gitterpunkte belegen, haben die Kodierungsbäume fixierte horizontale Positionen. Analog kann dies für die Kodierungsbäume unterhalb des Elementarpfades  $S(n, 1)$  gezeigt werden. Somit haben alle Kanten des Elementarbaumes  $S(n, 1)$  eine festgelegte Steigung, bis auf die Zentrumsverbindungen des Elementarpfades.

Für  $S(n, 1)$  müssen die Kodierungsbäume zwischen  $u_1$  und  $w_1$  gezeichnet werden. Für  $S(n, 2)$  wird dem  $\beta$ -Begrenzungsbaum in der Konstruktion der Knoten  $b_{11}$  an  $w_1$  angefügt. Durch die Konstruktion muss die Position rechts oberhalb von  $w_0$  durch  $b_{11}$  oder  $w_2$  belegt sein. Demzufolge haben die Kodierungsbäume festgelegte  $x$ -Koordinaten in  $\Gamma(S(n, 2))$ .

Im Folgenden wird angenommen, dass  $w_0$  auf den Koordinaten  $(0, 0)$  und  $b_{11}$  auf  $(1, 1)$  platziert sind. Im Konstruktionsschritt  $j \rightarrow j + 1$  ist die Induktionsannahme, dass die Koordinaten  $(j - \gamma, \gamma)$  mit  $\gamma \in \{1, \dots, j - 1\}$  durch die  $j - 1$  Knoten des  $j$ -ten Levels des Begrenzungsbaumes  $T_\beta$  belegt sind. Werden im Konstruktionsschritt  $j$  Knoten an diese Knoten des  $j$ -ten Levels angefügt, dann bleiben auf der Diagonalen  $(j + 1 - \gamma, \gamma)$  mit  $\gamma \in \{1, \dots, j\}$  nur  $j$  mögliche Koordinatenpositionen, die mit Kantenlänge eins erreicht werden können. Für den Fall, dass der  $w$ -Pfad  $(w_0, \dots, w'_{j+1})$  vertikal nach oben und  $(w_1, \dots, w_{j+1})$  ebenfalls vertikal nach oben eingebettet sind, muss die Kante  $(w_1, b_{11})$  horizontal eingebettet sein. Trotzdem sind die diagonalen Koordinaten dann belegt und die Argumentation kann analog durchgeführt werden. Somit sind die Koordinaten auf der Diagonalen immer besetzt. Letztendlich kann die Spalte  $C_n$  deshalb nicht nach rechts abknicken und ist somit fixiert. Für den  $d$ -Begrenzungsbaum kann dieselbe Argumentation durchgeführt werden. Die  $\alpha$ -Begrenzungsbaume haben auf dem 6-er Gitter dieselbe Eigenschaft, wie auf dem 8-er Gitter. Deshalb sind die  $x$ -Koordinaten der Kodierungsbäume in der Zeichnung  $\Gamma(S(n, j + 1))$  festgelegt.  $\square$

Diese Konstruktion stellt sicher, dass die 6-er Kodierungsbäume nicht nach außen abknicken können. O. B. d. A. sei der Elementarpfad in  $S(n, c + 1)$  horizontal eingebettet. Dann sind die Pfade  $(x_{i,c+1}, \dots, \bar{x}_{i,c+1})$  vertikal gezeichnet. Jedes  $5 \times 5$ -Quadrat eines 6-er Kodierungsbaumes belegt exakt zwei Gitterpunkte mehr als der Kodierungsbaum. Wie auf dem 8-er Gitter werden diese freien Gitterpunkte dazu verwendet, um den booleschen Ausdruck  $E$  in den Baum  $S(n, c + 1)$  zu kodieren.

Sei  $x_i$  eine Variable in der  $j$ -ten Klausel von  $E$  mit  $i \in \{1, \dots, n\}$  und  $j \in \{1, \dots, c\}$ . Wenn  $x_i$  in der  $j$ -ten Klausel nicht vorkommt, dann wird an den 6-er Kodierungsbaum von  $x_{ij}$  ein Striker  $(s_{ij}, s'_{ij}, s''_{ij})$  und an  $\bar{x}_{ij}$  ein Striker  $(\bar{s}_{ij}, \bar{s}'_{ij}, \bar{s}''_{ij})$  gehängt. Wenn  $x_i$  in der Klausel  $j$  nicht negiert auftritt, dann wird an den 6-er Kodierungsbaum  $x_i$  der Striker  $(s_{ij}, s'_{ij}, s''_{ij})$  angefügt. Schließlich wird für die negierte Variable in der  $j$ -ten Klausel der Striker  $(\bar{s}_{ij}, \bar{s}'_{ij}, \bar{s}''_{ij})$  an  $\bar{x}_{ij}$  angehängt. Diese Erweiterung von  $S(n, c + 1)$  wird im Folgenden  $T(E)$  genannt. Nach Lemma 3.11 sind in der Zeichnung  $\Gamma(T(E))$  mit Kantenlänge eins alle Kantenrichtungen bis auf die Kanten zwischen den vollen 6-er Bäumen des Elementarpfades und der zweiten Strikerkante  $(s'_{ij}, s''_{ij})$  festgelegt.

In der Zeichnung  $\Gamma(T(E))$  wird der Kodierungsbaum  $z_{ij} \in \{x_{ij}, \bar{x}_{ij}\}$  und der entsprechende Striker  $S = (s, s', s'')$  betrachtet. Aufgrund der nicht notwendigen Erhaltung der Reihenfolge der Nachbarknoten kann  $z_{ij}$  vertikal gespiegelt werden, so dass  $S$  entweder auf die linke Seite gezeichnet wird oder auf die rechte Seite. Nach Lemma 3.11 unterscheiden sich die  $y$ -Koordinaten der Spalten  $C_i$  und  $C_{i-1}$  oder  $C_{i+1}$  mit  $i \in \{2, \dots, n-1\}$  um maximal 1. Wenn es einen freien Gitterpunkt auf der rechten Seite von  $z_{i-1,j}$  oder entsprechend auf der linken Seite von  $z_{i+1,j}$  gibt, dann kann der Knoten  $s''$  auf diesem Gitterpunkt unter Berücksichtigung der Kantenlänge eins eingebettet werden. Als Beispiel siehe den Kodierungsbaum  $\bar{x}_{22}$  in Abbildung 3.46 mit dem Striker  $S$ , der nach rechts gezeichnet ist. Beachte, dass die Striker der Spalte  $C_1$  ( $C_n$ ) nur nach rechts (links) planar gezeichnet werden können.

**Lemma 3.12.** *Sei  $E$  ein boolescher Ausdruck in 3-KNF mit  $c$  Klauseln und  $n$  Variablen.  $E$  ist genau dann erfüllbar mit mindestens einem erfüllten und einem nicht-erfüllten Literal pro Klausel genau dann, wenn es eine Zeichnung  $\Gamma(T(E))$  mit Kantenlänge eins auf dem 6-er Gitter gibt.*

*Beweis.* “ $\Rightarrow$ ”: Sei  $\tau(E)$  eine erfüllende Belegung für den booleschen Ausdruck  $E$  mit  $n$  Variablen und  $c$  Klauseln. Der Baum  $T(E)$  wird, wie oben beschrieben, konstruiert. Um eine planare Zeichnung  $\Gamma(T(E))$  mit Kantenlänge eins zu erhalten, muss bestimmt werden, ob ein Pfad  $(x_{i1}, \dots, x_{i,c+1})$  in der oberen Hälfte und der entsprechende Pfad  $(\bar{x}_{i1}, \dots, \bar{x}_{i,c+1})$  in der unteren Hälfte eingebettet wird, oder umgekehrt. Wenn die Variable  $x_i$  mit  $i \in \{1, \dots, n\}$  mit *true* belegt ist, dann wird der entsprechende Pfad  $(x_{i1}, \dots, x_{i,c+1})$  in der oberen Hälfte eingebettet, andernfalls in der unteren Hälfte.

In der Zeile  $j$  oben (unten) für die Klausel  $j$  von  $\Gamma(T(E))$  gibt es ohne Striker  $n - 1$  freie Plätze, um entsprechend Striker planar einzubetten. Für die  $n - 3$  nicht vorhandenen Variablen in Klausel  $j$  werden nach der Konstruktion in der  $j$ -ten Zeile oben (unten)  $n - 3$  Striker eingefügt. Da die Belegung  $\tau(E)$  sicherstellt, dass jede

Klausel mindestens ein erfülltes und ein unerfülltes Literal besitzt, so müssen in jeder Zeile  $j$  oben (unten) mindestens ein Striker oder höchstens zwei Striker für vorhandene Variablen eingebettet sein. Das passt exakt zur Tatsache, dass in der  $j$ -ten Zeile in der oberen (unteren) Hälfte der Zeichnung für die  $j$ -te Klausel höchstens zwei Striker der entsprechenden Variablen der  $j$ -ten Klausel planar eingebettet werden können. Somit ist durch die gegebene erfüllte Belegung  $\tau(E)$  eine planare Zeichnung  $\Gamma(T(E))$  von  $T(E)$  mit Kantenlänge eins konstruiert worden.

“ $\Leftarrow$ ”: Sei  $\Gamma'(T(E))$  eine Zeichnung mit Kantenlänge eins von  $T(E)$ . Nach Lemma 3.11 sind die Steigungen aller Kanten festgelegt bis auf die verbindenden Kanten des Elementarpfades und der Kante  $(s', s'')$  der Striker. Ohne Striker gibt es, wie in “ $\Rightarrow$ ” bereits erwähnt,  $n - 1$  Löcher, d. h. zwei adjazente Gitterpunkte in der  $j$ -ten Zeile mit  $j \in \{1, \dots, c\}$  zwischen nebeneinander platzierten Kodierungsbäumen in der oberen Hälfte und ebenso  $n - 1$  Löcher in der unteren Hälfte. In jeder Zeile  $j$  (obere und untere Hälfte) sind  $n - 3$  Striker für nicht vorhandene Variablen der Klausel  $j$  angehängt. Deshalb können höchstens zwei Striker, die zu einer vorhandenen Variable in der aktuellen Klausel gehören, in der  $j$ -ten Zeile der oberen (unteren) Hälfte platziert werden. Für negierte und nicht-negierte Literale einer Klausel werden insgesamt drei Striker in den Baum eingefügt. Es ist nicht möglich, dass alle drei Striker in der oberen (unteren) Hälfte platziert werden. In einer planaren Zeichnung mit Kantenlänge eins müssen zwei Striker in der oberen und einer in der unteren Hälfte eingebettet sein, oder umgekehrt.

Ein Literal  $y_k$  mit  $k \in \{1, \dots, n\}$  ist entweder die Variable  $x_k$  oder seine Negation  $\bar{x}_k$ . Sei das Literal  $y_k$  in Klausel  $j$  vorhanden. Wenn  $y_k$  nicht negiert ist, dann wird das Literal auf erfüllt gesetzt, wenn der entsprechende Striker  $(s_{kj}, s'_{kj}, s''_{kj})$  in der oberen Hälfte der Zeichnung  $\Gamma(T(E))$  eingebettet ist. Andernfalls, wenn  $y_k$  negiert ist, dann ist das Literal nicht erfüllt, wenn der Striker  $(s_{kj}, s'_{kj}, s''_{kj})$  in der oberen Hälfte eingebettet ist. Somit erhält man eine erfüllende Belegung  $\tau'(E)$  hinsichtlich  $NAE3SAT$  mit mindestens einem erfüllten Literal und mindestens einem nicht erfüllten Literal in jeder Klausel.  $\square$

**Satz 3.22.** *Sei  $T$  ein Baum mit Grad sechs. Es ist  $\mathcal{NP}$ -hart zu entscheiden, ob eine  $U_6$ -Zeichnung von  $T$  mit Kantenlänge eins existiert.*

**$UL_k$ -,  $UP_k$ - und  $ULP_k$ -Zeichnungen** Im Folgenden werden  $U_k$ -Zeichnungen durch verschiedene Ästhetikkriterien eingeschränkt und dann die Komplexität untersucht. Dabei werden Kriterien, wie lokale Uniformität oder/und Pattern verwendet, um dann  $UL_k$ -,  $UP_k$ - und  $ULP_k$ -Zeichnungen zu erhalten. Es ist zu beachten, dass diese Ästhetikkriterien nur auf gewurzelte Bäume angewendet werden können, siehe Definition 3.9 und 3.7.

**Satz 3.23.** *Sei  $T$  ein  $d$ -ärer Baum mit  $k \in \{4, 6, 8\}$  und  $d < k$ , dann ist es  $\mathcal{NP}$ -hart zu entscheiden, ob es eine  $UL_k$ -Zeichnung von  $T$  mit Kantenlänge eins gibt.*

*Beweis.* Für  $UL_k$ -Zeichnungen mit  $k \in \{4, 6, 8\}$  wird dieselbe Konstruktion entsprechend  $k \in \{4, 6, 8\}$  angewendet. Die resultierenden Zeichnungen sind dann durch

die lokale Uniformität und die einheitlichen Kantenlängen dieselben Zeichnungen wie für  $U_k$ -Zeichnungen.  $\square$

Betrachtet man  $UP_k$ - und  $ULP_k$ -Zeichnungen, müssen die ausgehenden Kanten eines Knotens den Pattern folgen. Für das 4-er und 8-er Gitter können die Ergebnisse leicht transformiert übernommen werden.

**Satz 3.24.** *Sei  $T$  ein  $d$ -ärer Baum mit  $k \in \{4, 8\}$  und  $d < k$ , dann ist es  $\mathcal{NP}$ -hart zu entscheiden, ob es eine  $UP_k$ - oder  $ULP_k$ -Zeichnung von  $T$  mit Kantenlänge eins gibt.*

*Beweis.* Die Konstruktion von Bhatt und Cosmadakis auf dem 4-er Gitter erzwingt auf dem 4-er Gitter bereits  $UP_4$ - bzw.  $ULP_4$ -Zeichnungen.

Für  $k = 8$  werden aufgrund der Pattern-Eigenschaft die Kanten zwischen den vollen 8-er Bäumen auf dem Elementarpfad horizontal gezeichnet. Damit wird ebenfalls eine  $UP_8$ -Zeichnung erreicht und der Beweis kann analog zum Beweis von Lemma 3.9, mit der beschriebenen Reduktion durchgeführt werden.  $\square$

Die Ergebnisse auf dem 4-er und dem 8-er Gitter für  $U_k$ -Zeichnungen ( $k \in \{4, 8\}$ ) können somit einfach auf Pattern-Zeichnungen übertragen werden. Für das 6-er Gitter ist dies nicht so einfach möglich, da durch die Pattern die Zeichnungen der vollen 6-er Bäume und die 6-er Kodierungsbäume nicht mehr die notwendigen Eigenschaften des Beweises haben. Betrachte dazu Abbildung 3.45(a) mit der darin gegebenen Beschriftung. Diese Zeichnung ist keine Pattern-Zeichnung, denn dazu müsste eine der beiden Kanten zu den Blättern von  $t_1$  diagonal nach unten gezeichnet sein, siehe Abbildungen 3.15(c) und 3.16(c). Somit müssten die Knoten  $s_1$  und  $t_1$  vertauscht werden, um eine Zeichnung mit Kantenlänge eins zu ermöglichen. Damit wäre  $s_1$  nicht mehr am Rand des  $5 \times 5$  Quadrats, so dass benachbarte volle 6-er Bäume nicht mehr mit Kantenlänge eins angeschlossen werden können. Somit kann die Konstruktion nicht auf einfache Weise angepasst werden. Trotzdem wird folgende Vermutung angestellt.

**Vermutung 3.2.** *Sei  $T$  ein Baum mit Grad sechs. Es ist  $\mathcal{NP}$ -hart zu entscheiden, ob eine  $UP_6$ - oder  $ULP_6$  Zeichnung von  $T$  mit Kantenlänge eins auf dem 6-er Gitter existiert.*

Bisher wurde für nicht-ordnungserhaltende Zeichnungen die Komplexität der einheitlichen Kantenlänge betrachtet. Es ist jedoch auch von Interesse, die Komplexität der minimalen Fläche zu bestimmen. Dieses Problem wird im folgenden Abschnitt für das 8-er Gitter behandelt.

### 3.2.3.3 Minimale Fläche

In diesem Abschnitt liegt das Interesse nicht mehr auf den einheitlichen Kantenlängen in einer Zeichnung, sondern auf der benötigten Fläche des kleinsten umschließenden Rechtecks. Für das 4-er Gitter wurde die  $\mathcal{NP}$ -Härte des Bestimmens der minimalen Fläche festgestellt [Bra90].

**Vermutung 3.3.** *Sei  $T$  ein Baum mit Grad acht und  $A > 1$ . Es ist  $\mathcal{NP}$ -hart zu entscheiden, ob eine  $U_8$ -Zeichnung  $\Gamma(T)$  innerhalb der Fläche  $A$  gezeichnet werden kann.*

*Beweis.* (Idee). Es wird, wie in Lemma 3.9, von  $NAE3SAT$  reduziert. Sei  $E$  ein boolescher Ausdruck mit  $c$  Klauseln und  $n$  Variablen und sei der Baum  $T(E)$  vom Grad acht konstruiert. Dazu werden zusätzliche Blätter eingefügt. Sei  $j \in \{1, \dots, c\}$ . Für jeden Kodierungsbaum  $x_{1j}$  und  $\bar{x}_{1j}$  der ersten Spalte  $C_1$  wird eine Kante  $(s_{1j}, b_{1j})$  und  $(\bar{s}_{1j}, \bar{b}_{1j})$  eingefügt. Für die  $n$ -te Spalte wird das Gleiche in analoger Weise ausgeführt. In Abbildung 3.47 sind die zusätzlichen Strikerknoten gezeigt. Siehe dazu in der ersten Spalte beispielsweise die Kodierungsbäume  $x_{11}$  bis  $x_{13}$  und letzten  $\bar{x}_{41}$  bis  $\bar{x}_{43}$ .

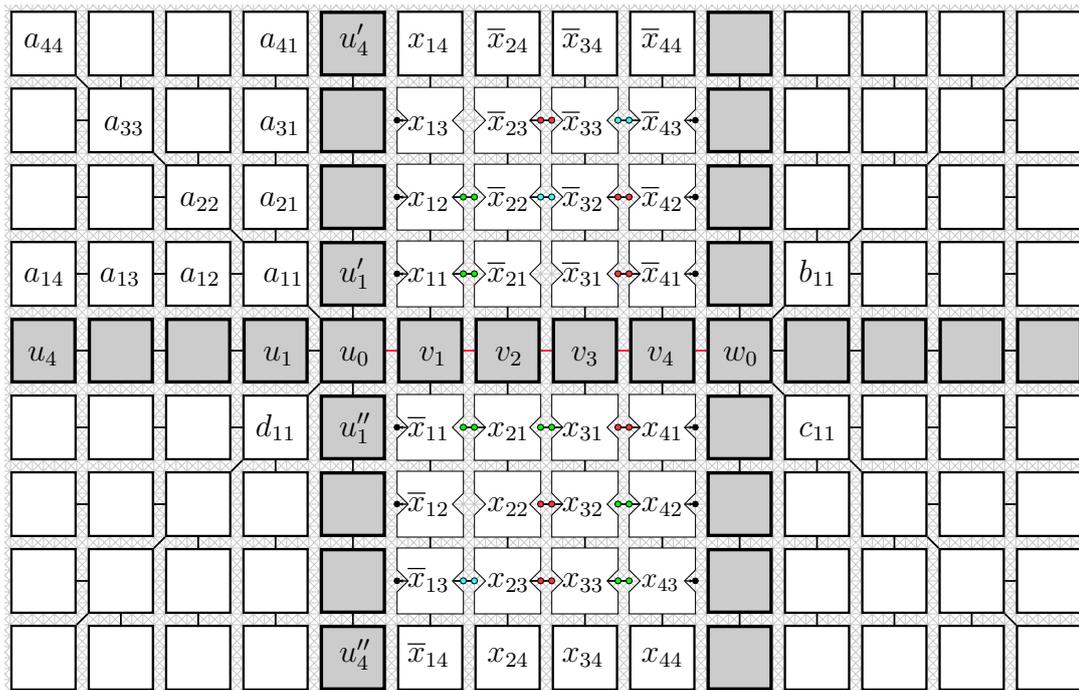
Sei  $A = W \cdot H$  mit  $W = 7(2(c+1) + n + 2)$  und  $H = 7(2(c+1) + 1)$ . Der boolesche Ausdruck  $E$  ist erfüllbar mit mindestens einem erfüllten und einem nicht-erfüllten Literal pro Klausel genau dann, wenn es eine  $U_8$ -Zeichnung innerhalb der Fläche  $A$  von  $T(E)$  gibt.

“ $\Rightarrow$ ”: Es wird ähnlich zum „ $\Rightarrow$ “-Teil im Beweis von Lemma 3.9 argumentiert. Der Elementarpfad  $(u_0, v_1, \dots, v_n, w_0)$  wird horizontal nach rechts gezeichnet mit identischen  $y$ -Koordinaten. Die zu den Strikerblättern hinzugefügten Kanten werden in  $C_1$  ( $C_n$ ) alle nach links (rechts) ausgerichtet. Unter Verwendung der Belegung von  $\tau(E)$  werden die üblichen Striker, wie in Lemma 3.9 ausgerichtet. Dann hat  $\Gamma(T(E))$  eine Höhe  $H$ . Die gesamte Breite hängt von der Summe der Länge eines  $u$ -Pfades, dem Elementarpfad und dem  $w$ -Pfad ab. Diese Summe ergibt  $W$ .

“ $\Leftarrow$ ”: Sei  $\Gamma(T(E))$  eine Zeichnung von  $T(E)$  innerhalb der Fläche  $A$ . Die Anzahl der verfügbaren Gitterpunkte der Fläche ist  $A = H \cdot W$  mit  $H = 7(2(c+1) + 1)$  und  $W = 7(2(c+1) + n + 2)$ . Die Anzahl der Knoten in  $T(E)$  ist um  $2c$  kleiner als die Anzahl der verfügbaren Gitterpunkte in  $A$ . Für jede Klausel  $j$  werden exakt zwei Gitterpunkte frei gelassen, was die Zeichnung einer Zeile in der oberen oder der unteren Hälfte und somit die ganze Zeichnung stark festziehen soll, so dass man die Belegung  $\tau(E)$  genauso wie in Lemma 3.9 ablesen kann.  $\square$

**Vermutung 3.4.** *Sei  $T$  ein 7-närer Baum und  $A > 1$ . Es ist  $\mathcal{NP}$ -hart zu entscheiden, ob es eine  $UP_8$ -, eine  $UL_8$ - oder eine  $ULP_8$ -Zeichnung innerhalb der Fläche  $A$  gibt.*

*Beweis.* Zuerst werden  $ULP_8$ -Zeichnungen betrachtet. Sei  $\Gamma(T(E))$  lokal uniforme Pattern-Zeichnungen innerhalb der Fläche  $A = H \cdot W$ , die identisch zum Beweis von Vermutung 3.3 ist. Da die Zeichnung nur Kanten mit Länge eins besitzt, ist sie bereits lokal uniform. Weil die Höhe erzwingt, dass die den Elementarpfad verbindenden Kanten horizontal gezeichnet werden, handelt es sich bei der Zeichnung bereits um eine  $ULP_8$ -Zeichnung. Jede  $ULP_8$ -Zeichnung ist bereits eine  $UL_8$ - oder  $UP_8$ -Zeichnung, so dass für letztere die gleichen Argumente wie für  $ULP_8$ -Zeichnungen Verwendung finden.  $\square$



**Abbildung 3.47.**  $T(E)$  von  $E = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$  ( $n = 4$  und  $r = 3$ ) und Belegung  $x_1 = true, x_2 = x_3 = x_4 = false$

Für Zeichnungen von 5-nären Bäumen auf dem 6-er Gitter kann die oben beschriebene Konstruktion so nicht einfach angewendet werden, da die Zeichnung dadurch nicht so „festgezurr“ ist, dass dieselben Argumente angewendet werden können.

# 4

## Zeichenverfahren auf $k$ -Gitter

---

Nach den Abschätzungen für die Fläche und die Komplexität von vollständigen  $d$ -nären Bäumen, geordneten und ungeordneten  $d$ -nären Bäumen mit  $k \in \{4, 6, 8\}$  und  $d < k$ , werden in diesem Kapitel Verfahren zum Zeichnen von geordneten  $d$ -nären Bäumen auf Gittern vorgestellt. Diese verwenden bekannte Ansätze aus hierarchischen Verfahren und erweitern diese. Außerdem werden die eingeführten Ansätze und die dazu entwickelten Ästhetikkriterien zum Sicherstellen der Qualität der Zeichnungen gefordert. Die Heuristiken zum Berechnen von ordnungserhaltenden lokal uniformen Pattern-Zeichnungen von  $d$ -nären Bäumen für  $k \in \{4, 6, 8\}$  und  $d < k$  vorgestellt. Dazu werden die im hierarchischen Fall verwendeten Ästhetikkriterien auf das Gitter transformiert. Die Heuristiken verwenden eine erweiterte Form der Konturen, die ähnlich bereits zum Zeichnen von hierarchischen Zeichnungen eingeführt wurden [RT81, Wal90, BJL02]. Die Verfahren können ebenfalls für nicht lokal-uniforme und nicht-Pattern-Zeichnungen verwendet werden, um die Fläche von gegebenen Zeichnungen zu verringern.

### 4.1 Zeichnungen von $d$ -nären Bäumen mit Unterbaum Separation

Im Folgenden wird eine Heuristik zum Berechnen von ordnungserhaltenden lokal-uniformen Pattern-Zeichnungen, den  $OLP_k$ -Zeichnungen, von  $d$ -nären Bäumen für  $k \in \{4, 6, 8\}$  und  $d < k$  vorgestellt. Dazu werden die im hierarchischen Fall einge-

fürten Ästhetikkriterien  $AK1 - AK4$  auf das  $k$ -Gitter transformiert.

Das Ästhetikkriterium Knoten desselben Levels auf einer Geraden ( $AK1$ , Definition 3.2) aus dem hierarchischen Fall wird ersetzt durch die lokale Uniformität auf dem  $k$ -Gitter (siehe Definition 3.9). Somit haben alle Kinder denselben Abstand zum entsprechenden Vaterknoten. Diese Forderung lehnt sich ebenfalls der Forderung von Bloesch an, dass im hierarchischen Fall mit rechteckigen Knoten die Oberkante auf einer Linie platziert sein sollen [Blo93].

Linkes Kind links und rechtes Kind rechts vom Vater ( $AK2$ , Definition 3.3) wird dahingehend übernommen, dass die Einbettung des vorgegebenen Baumes eingehalten wird und zusätzlich die Pattern beachtet werden, siehe Definition 3.7.

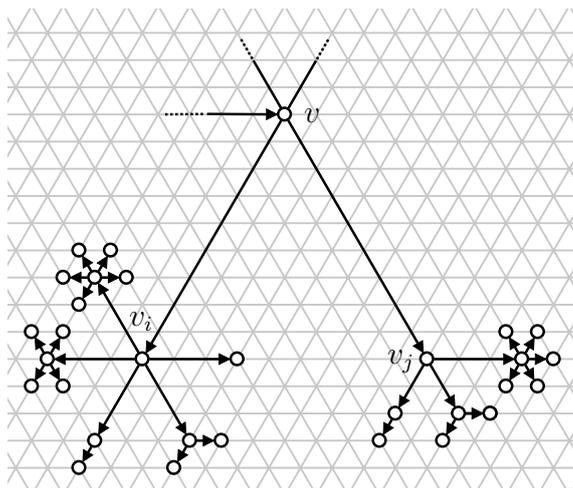
Die Forderung Vater zentriert über den Kindern ( $AK3$ , Definition 3.4) wird durch die Symmetrie der Pattern und die lokale Uniformität sichergestellt. Dies erzwingt, dass die ausgehenden Kanten symmetrisch zur eingehenden Kante von  $v$  gezeichnet werden. Damit schließen die ausgehenden Kanten von  $v$  zum linkesten Kind und zum rechtesten Kind die Winkel  $\varphi$  und  $-\varphi$  ein.

Das Ästhetikkriterium isomorphe Unterbäume bis auf Verschiebung gleich zu zeichnen ( $AK4$ , Definition 3.6) wird ähnlich zum hierarchischen Fall erfüllt. Der einzige Unterschied ist, dass diese zusätzlich rotiert werden dürfen. Ebenfalls sollen Unterbäume und deren Spiegelungen bis auf die Spiegelung identisch gezeichnet werden, um gleiche Strukturen leichter in der Zeichnung wiederzufinden.

Die Kriterien der ganzzahligen Koordinaten für Knoten, Planarität und ein gewährleisteter Mindestabstand zwischen Knoten werden ohne Änderung übernommen. Aus den Anpassungen ergeben sich als Ästhetikkriterien die Ordnungserhaltung, lokale Uniformität und das Zeichnen nach den Pattern. Im Hinblick auf die Rechteck Unterbaum Separation wird in diesem Abschnitt eine andere Form von Unterbaum Separation eingeführt, welche in Abschnitt 4.1.1 definiert wird.

Wie auch in den bereits diskutierten hierarchischen Algorithmen von Reingold [RT81] und Tilford oder Walker [Wal90] wird rekursiv überprüft, wie weit Unterbäume zusammengeschoben werden können, vgl. Abschnitt 3.1.1. Durch die angepassten Ästhetikkriterien ist dies jedoch nur noch durch den Freiheitsgrad der Kantenlängen realisierbar.

Betrachte dazu das Beispiel in Abbildung 4.1, wo die ausgehenden Kanten aller Knoten gleich lang sind und die Eigenschaft einer Pattern-Zeichnung erfüllen. Dies gilt insbesondere auch für die ausgehenden Kanten von der Wurzel  $v$ , des gezeigten Unterbaumes. Betrachte die ausgehenden Kanten  $(v, v_i)$  und  $(v, v_j)$  des Knotens  $v$ . Es ist möglich die Kanten  $(v, v_i)$  und  $(v, v_j)$  simultan um vier zu verkürzen und dabei die geforderten Ästhetikkriterien zu erfüllen und damit die Fläche zu verringern. Die Heuristik zielt darauf ab, die ausgehenden Kanten eines Knotens unter Beibehaltung der Planarität simultan zu verkürzen, um die lokale Uniformität zu erhalten. Dies wird in einem Bottom-Up Ansatz verwirklicht und ist angelehnt an die bereits vorgestellten hierarchischen Verfahren. Die Berechnung der Verkürzung der Kanten im Unterbaum  $T_v$  verläuft rekursiv. Die dazu verwendeten Datenstrukturen werden nachfolgend definiert. In jedem Rekursionsschritt werden Abstände zwischen Kindunterbäumen berechnet, um den möglichen Verkürzungswert zu bestimmen. Dieser



**Abbildung 4.1.** Ausschnitt eines Baumes auf dem 6-er Gitter mit Knoten  $v$  und den Kindern  $v_i$  und  $v_j$

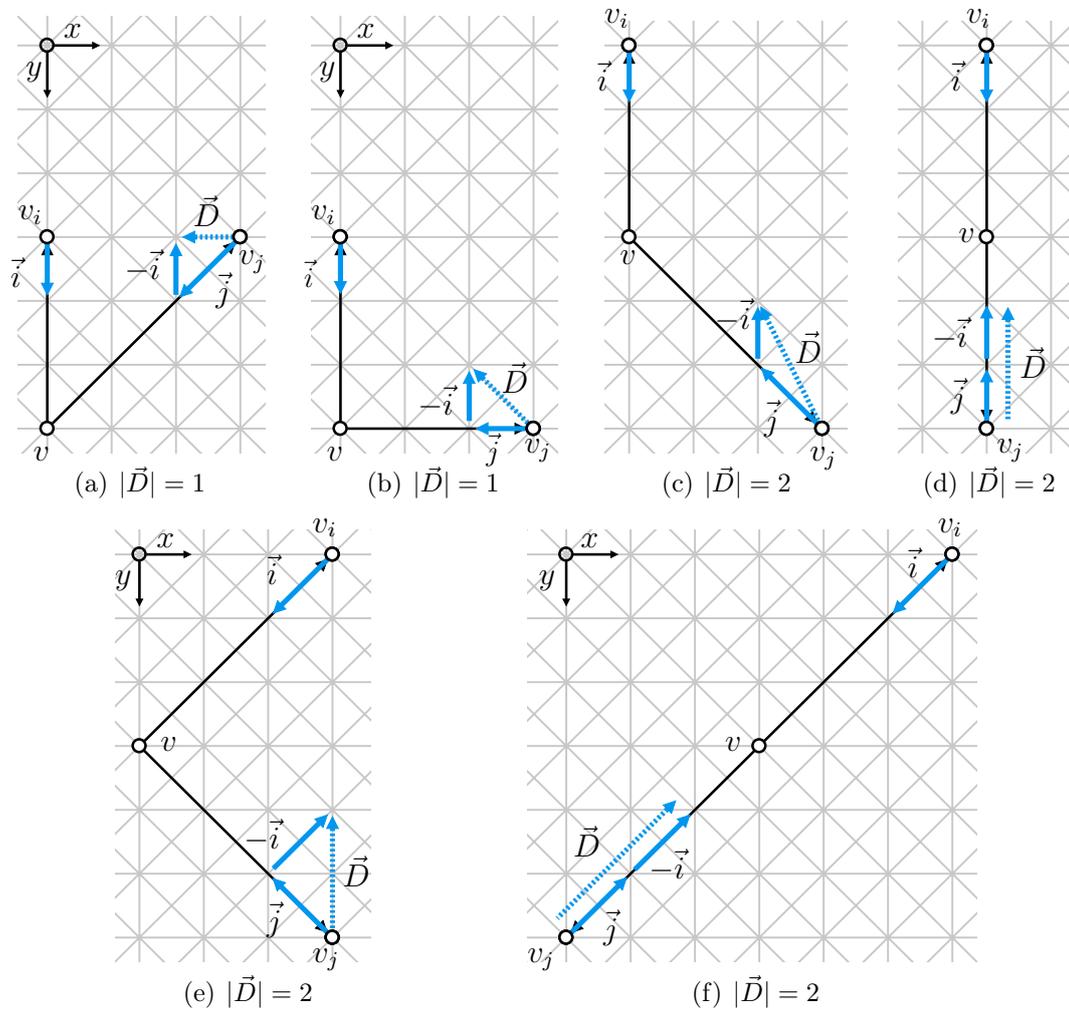
Wert gibt an, um welchen Wert die ausgehenden Kanten des Knotens  $v$  verkürzt werden können.

### 4.1.1 Definitionen

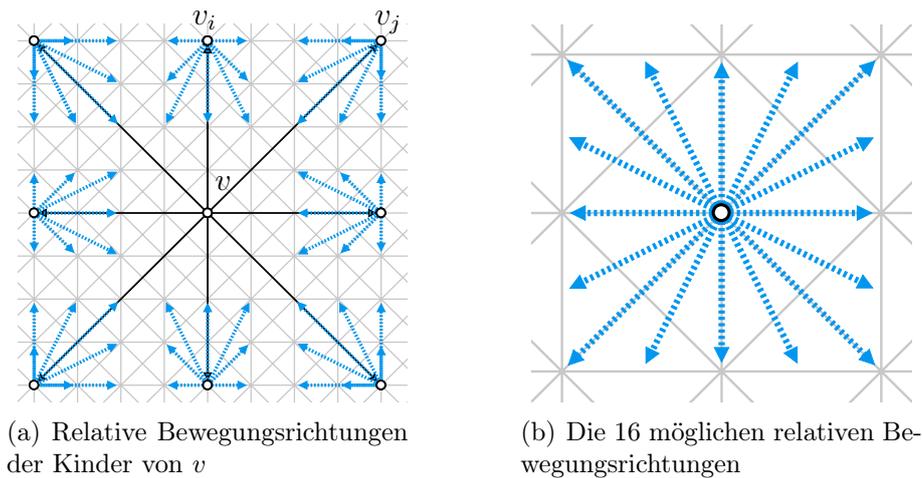
Um das Verfahren beschreiben zu können, werden anfangs einige Definitionen eingeführt. Des Weiteren werden Eigenschaften erläutert, die im nachfolgend beschriebenen Verfahren von Bedeutung sind.

Gegeben sei eine planare Zeichnung  $\Gamma_k(T)$  eines  $d$ -nären Baumes auf dem  $k$ -Gitter mit  $k \in \{4, 6, 8\}$  und  $d < k$ . Werden die ausgehenden Kanten eines Knotens  $v$  zu den Kindern  $v_1, \dots, v_l$  mit  $l \in \{1, \dots, d\}$  verkürzt, so bewegt sich jeder Kindknoten  $v_i$  relativ zu einem seiner Geschwisterknoten  $v_j$  ( $i \neq j$  und  $i \in \{v_1, \dots, v_l\}$ ), siehe Abbildung 4.2. Darin sind die verschiedenen Fälle gezeigt, wie sich Geschwisterknoten  $v_i$  und  $v_j$  zueinander bewegen können, wenn die Kanten zum Vaterknoten  $v$  um eins verkürzt werden. Dabei geben die Vektoren  $\vec{D}_i$  und  $\vec{D}_j$  an, in welche Richtung sich der entsprechende Kindknoten bei der Verkürzung verschiebt. In Abbildung 4.2(a) ist  $\vec{D}_i = (0, 1)$  und  $\vec{D}_j = (-1, 1)$  gezeigt. Die Differenz  $\vec{D}_{i,j} = \vec{D}_j - \vec{D}_i = (-1, 0)$  gibt an, wie sich  $v_j$  relativ zu  $v_i$  bewegt. Im Folgenden wird diese die *relative Bewegungsrichtung*  $\vec{D}_{i,j}$  von  $v_j$  zu  $v_i$  genannt. Ist es im Folgenden ersichtlich, welche Knoten die relative Bewegungsrichtung betreffen, so werden die Indizes weggelassen. Die restlichen relativen Bewegungsrichtungen auf dem 8-er Gitter werden analog berechnet, siehe Abbildungen 4.2(b), 4.2(c) und 4.2(d). Für andere Richtungen können die übrigen Fälle auf dem 8-er Gitter symmetrisch zu den in Abbildung 4.2 gezeigten bestimmt werden, siehe Abbildung 4.3(a).

Betrachtet man das 6-er Gitter, so wird die Menge der relativen Bewegungsrichtungen analog festgelegt, siehe Abbildung 4.4. Dazu ist erst das asymmetrische



**Abbildung 4.2.** Relative Bewegungsrichtung von  $v_j$  in Abhängigkeit von  $v_i$  und der entsprechende Verkürzungsfaktor  $|\vec{D}|$



**Abbildung 4.3.** Relative Bewegungsrichtungen auf dem 8-er Gitter

6-Gitter gezeigt, in dem im Vergleich zum 8-er Gitter eine diagonale Achse weggelassen wird. Die sich aus Abbildung 4.4(a) ergebenden relativen Bewegungsrichtungen sind in Abbildung 4.4(b) dargestellt. Die Menge der Bewegungsrichtungen auf dem 6-er Gitter sind eine echte Teilmenge der Bewegungsrichtungen auf dem 8-er Gitter.

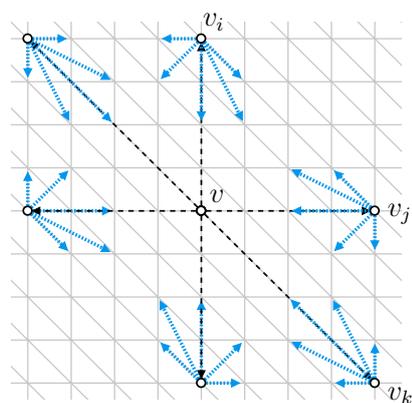
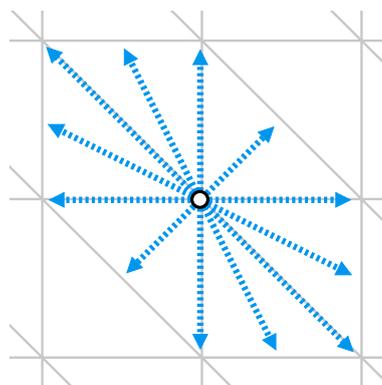
Dasselbe kann für das 4-er Gitter durchgeführt werden, wobei sich dann ebenfalls eine echte Teilmenge an relativen Bewegungsrichtungen ergibt, siehe Abbildung 4.5.

Nach der relativen Bewegungsrichtung  $\vec{D}_{i,j}$  ist es ebenfalls notwendig, zu bestimmen, wie sich der Abstand zwischen  $v_i$  und  $v_j$  bei einer Verkürzung um eins entlang  $\vec{D}_{i,j}$  ändert. In den Fällen zu Abbildungen 4.2(a) und 4.2(b) verringert sich bei einer Verkürzung um eins der Abstand um eins. Die maximale Verkürzung der ausgehenden Kanten von  $v$  ist unter anderem bestimmt durch den Abstand  $d_{\vec{D}_{i,j}}(v_i, v_j)$  entlang  $\vec{D}_{i,j}$  minus eins. Da sich die Unterbäume von  $v_i$  und  $v_j$  in Abbildungen 4.2(c), 4.2(d), 4.2(e) und 4.2(f) um  $|\vec{D}_{i,j}| = 2$  pro Verkürzung um eins annähern, können die ausgehenden Kanten von  $v$  nur um maximal  $\frac{1}{2} \cdot (d_{\vec{D}_{i,j}}(v_i, v_j) - 1)$  verkürzt werden ohne die Planarität zu zerstören. In Abbildung 4.3(a) sind alle Fälle dargestellt und in Abbildung 4.3(b) sind alle möglichen relative Bewegungsrichtungen für das 8-er Gitter zusammengefasst.

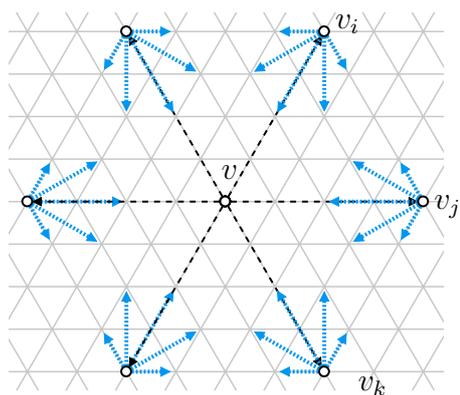
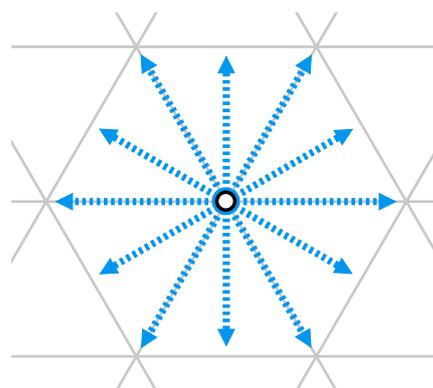
**Definition 4.1.** Die relative Bewegungsrichtung  $\vec{D}$  gibt an in welche Richtung sich zwei Geschwisterkonturen bei einer Verkürzung aufeinander zu bewegen, und um welchen Faktor  $|\vec{D}|$  sich der Abstand, relativ zum Verkürzungswert, ändert.

Es ist zu beachten, dass es auf jedem Gitter exakt doppelt so viele relative Bewegungsrichtungen gibt wie Gitterrichtungen. Vier Gitterrichtungen auf dem 4-er Gitter haben 8 relative Bewegungsrichtungen zur Folge.

**Behauptung 4.1.** Auf dem  $k$ -Gitter mit  $k \in \{4, 6, 8\}$  gibt es  $2k$  relative Bewegungsrichtungen.

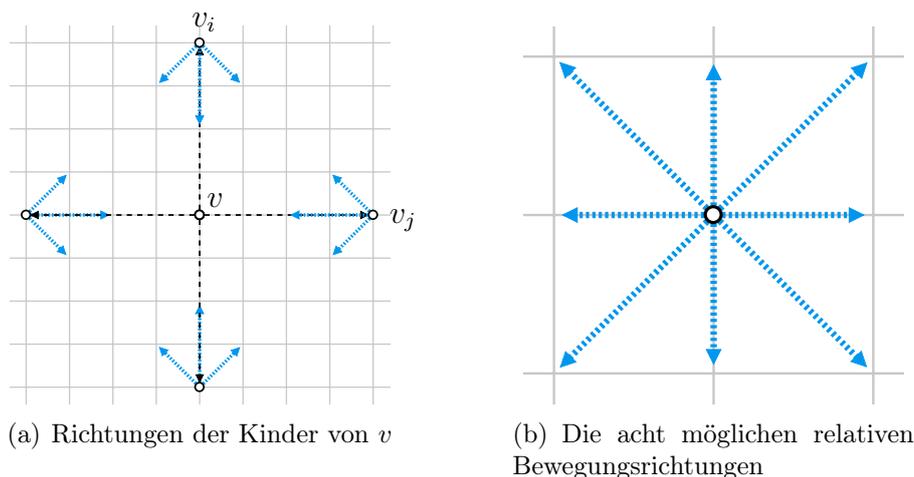
(a) Richtungen der Kinder von  $v$ 

(b) Die zwölf möglichen relativen Bewegungsrichtungen

(c) Richtungen der Kinder von  $v$ 

(d) Die zwölf möglichen relativen Bewegungsrichtungen

**Abbildung 4.4.** Bewegungsrichtungen auf dem symmetrischen und asymmetrischen 6-er Gitter



**Abbildung 4.5.** Bewegungsrichtungen auf dem 4-er Gitter

Aus Behauptung 4.1 und den vorhergehenden Eigenschaften ergibt sich folgende Definition.

**Definition 4.2 (Richtungsvektor).** Ein Richtungsvektor wird durch einen zwei-dimensionalen Vektor  $\vec{D} = (D_x, D_y) \in \mathbb{Z}^2$  festgelegt.  $\mathcal{D}_k$  ist Menge von Richtungsvektoren im  $k$ -Gitter mit  $2k$  Richtungsvektoren. Dann gilt

$$\mathcal{D}_4 = \left\{ \begin{pmatrix} 0 \\ \pm 1 \end{pmatrix}, \begin{pmatrix} \pm 1 \\ \pm 1 \end{pmatrix}, \begin{pmatrix} \pm 1 \\ 0 \end{pmatrix} \right\}, \quad (4.1)$$

$$\mathcal{D}_6 = \mathcal{D}_4 \cup \left\{ \pm \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}, \pm \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} \right\}, \quad (4.2)$$

$$\mathcal{D}_8 = \mathcal{D}_6 \cup \left\{ \pm \begin{pmatrix} 0.5 \\ -1 \end{pmatrix}, \pm \begin{pmatrix} 1 \\ -0.5 \end{pmatrix} \right\}. \quad (4.3)$$

Für einen Vektor  $\vec{D} = \begin{pmatrix} D_x \\ D_y \end{pmatrix} \in \mathcal{D}_k$  wird der dazu orthogonale Vektor  $\vec{D}^\circ = \begin{pmatrix} -D_y \\ D_x \end{pmatrix} \in \mathcal{D}_k$  definiert. Ein durch eine gerichtete Kante  $e = (u, v)$  mit  $u = (x_u, y_u)$

und  $v = (x_v, y_v)$  induzierter Richtungsvektor ist  $\vec{e} = \begin{pmatrix} x_v - x_u \\ y_v - y_u \end{pmatrix}$ . In einem Baum

$T$  ist der induzierte Richtungsvektor  $\vec{v}$  von  $v$  der induzierte Richtungsvektor der eingehenden Kante von  $v$ .

Die Richtungen der Richtungsvektoren aus  $\mathcal{D}_k$  mit  $k \in \{4, 6, 8\}$  und  $|\mathcal{D}_k| = 2k$  stimmen exakt mit den Richtungen in Abbildung 4.3(b), Abbildung 4.4(d), Abbildung 4.4(b) und Abbildung 4.5(b) überein. Bis zu diesem Zeitpunkt sind die Richtungen bekannt, nach denen sich die Unterbäume der Geschwisterknoten aufeinander zu bewegen, wenn sich die entsprechenden eingehenden Kanten simultan verkürzen.

**Definition 4.3 (Richtungsabstand).** Sei  $S$  ein Segment einer Kante,  $x$  ein Punkt und  $\vec{D} \in \mathcal{D}$  ein Richtungsvektor. Dann ist der Richtungsabstand  $d_{\vec{D}}(x, S)$  die Länge des Segmentes parallel zu  $\vec{D}$  mit einem Endpunkt auf  $x$  und der andere Endpunkt auf  $S$ . Wenn so ein Segment nicht existiert, dann ist  $d_{\vec{D}}(x, S) = \infty$ .

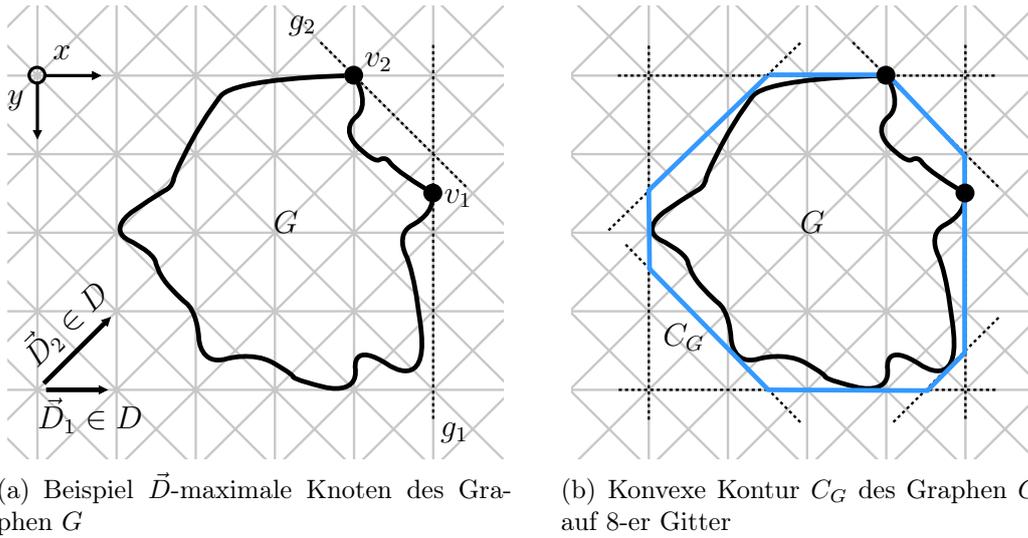
Die Distanz zwischen zwei Koordinaten kann ebenfalls durch andere Metriken, wie z. B. der euklidischen Metrik, bestimmt werden. Ein Nachteil der Maximumsmetrik in Bezug auf die euklidische Metrik ist beispielsweise, dass die Maximumsmetrik den diagonalen Abstand auf dem 8-er Gitter verzerrt. Die Maximumsmetrik legt den diagonalen Abstand zwischen den Koordinaten  $(x, y)$  und  $(x+1, y+1)$  mit eins fest, wobei die euklidische Metrik  $d_{\text{eukl}}((x, y), (x+1, y+1)) = \sqrt{2}$  die gezeichnete Länge der Kante exakt wiedergibt. Somit beachtet die euklidische Metrik die Symmetrie und die wirklichen Kantenlängen in der Ebene. Der entscheidende Nachteil der euklidischen Metrik ist jedoch, dass sie irrationale Zahlen in den Berechnungen verwendet, was in automatischen Berechnungen zu Rundungsfehlern führt. Somit ist es bezüglich Ästhetik und Symmetrie zwar ein Nachteil, sich auf ganzzahlige Koordinaten zu beschränken, aber in den Berechnungen treten keine Ungenauigkeiten im Sinne von Rundungsfehlern auf. Somit wird weiterhin die Maximumsmetrik zur Abstandsbestimmung verwendet. Es ist anzumerken, dass dieser Nachteil nur auf dem 8-er Gitter auftritt. Auf dem 4-er und 6-er Gitter liefern Maximumsmetrik und euklidische Metrik dieselben Ergebnisse.

**Definition 4.4 ( $\vec{D}$ -maximaler Knoten,  $\vec{D}$ -maximale Gerade).** Sei der Richtungsvektor  $\vec{D} = \begin{pmatrix} x_{\vec{D}} \\ y_{\vec{D}} \end{pmatrix} \in \mathcal{D}$  eine Richtung und  $\Gamma(G)$  eine Zeichnung eines Graphen  $G = (V, E)$ . Ein Knoten  $u \in V$  auf Koordinaten  $(x_u, y_u)$  ist  $\vec{D}$ -maximal, wenn gilt

$$\begin{pmatrix} x_u \\ y_u \end{pmatrix} \circ \vec{D} = \max \left\{ \begin{pmatrix} x_v \\ y_v \end{pmatrix} \circ \begin{pmatrix} x_{\vec{D}} \\ y_{\vec{D}} \end{pmatrix} \mid v \in G \right\}.$$

Die Operation  $\circ$  definiert das Skalarprodukt zwischen zwei Vektoren. Die Gerade  $g$  durch einen  $\vec{D}$ -maximalen Knoten  $u$  mit der zu  $\vec{D}$  orthogonalen Richtung  $\vec{D}^\circ$  wird  $\vec{D}$ -maximale Gerade genannt.

Sei  $\Gamma(G)$  eine Zeichnung eines Graphens  $G$  und  $v \in V$  mit den Koordinaten  $(x_v, y_v)$ . Dann ist  $v$  ein  $\vec{D}$ -maximaler Knoten, wenn er am weitesten in Richtung  $\vec{D}$

(a) Beispiel  $\vec{D}$ -maximale Knoten des Graphen  $G$ (b) Konvexe Kontur  $C_G$  des Graphen  $G$  auf 8-er Gitter**Abbildung 4.6.**  $\vec{D}$ -maximale Knoten und konvexe Kontur einer Zeichnung  $\Gamma(G)$  auf dem 8-er Gitter

platziert ist. Für die Richtung  $\vec{D} = (1, 0)$ , d. h.  $\vec{D}$  zeigt horizontal nach rechts, ist das ein Knoten, der in  $\Gamma(G)$  am weitesten rechts platziert ist, wie z. B. der Knoten  $v_1$  der Zeichnung  $\Gamma(G)$  des Graphen  $G$  in Abbildung 4.6(a). Der Knoten  $v_2$  ist  $\vec{D}$ -maximal in Richtung  $\vec{D} = (1, -1)$ .  $\vec{D}$ -maximale Geraden in der Zeichnung  $\Gamma(G)$  von  $\vec{D}_1$  und  $\vec{D}_2$  sind  $g_1$  und  $g_2$ .

**Definition 4.5 (Konvexe Kontur).** Gegeben sei eine Zeichnung  $\Gamma(G)$  eines Graphen  $G$ . Sei  $\vec{D} \in \mathcal{D}_k$  und  $g_{\vec{D}}$  eine  $\vec{D}$ -maximale Gerade, die eine Ebene  $\mathcal{E}$  in jeweils zwei Halbebenen unterteilt.  $\mathcal{E}_{G,\vec{D}}$  ist dabei die abgeschlossene Halbebene in der Knoten von  $G$  platziert sind und  $\mathcal{E}_{-G,\vec{D}}$  die offene Halbebene, in der sich keine Knoten befinden. Die offene Halbebene  $\mathcal{E}_{-G,\vec{D}}$  ist festgelegt durch  $\mathcal{E} \setminus \mathcal{E}_{G,\vec{D}}$ . Der Durchschnitt  $F_G$  der abgeschlossenen Halbebenen  $\mathcal{E}_{G,\vec{D}}$  für  $\vec{D} \in \mathcal{D}_k$  ist definiert durch

$$F_G = \bigcap_{\vec{D} \in \mathcal{D}_k} \{ \mathcal{E}_{G,\vec{D}} \}.$$

Eine konvexe Kontur  $C_G$  ist der Rand der Fläche  $F_G$ .

Für ein  $k$ -Gitter ist eine konvexe Kontur  $C_G$  ein Polygon mit höchstens  $k$  Ecken, siehe als Beispiel auf dem 8-er Gitter Abbildung 4.6(b). Für einen einzelnen Knoten  $v$  mit Koordinaten  $(x_v, y_v)$  besteht die *triviale konvexe Kontur* somit aus einem Punkt, da sich alle  $k$   $\vec{D}$ -maximalen Geraden im Punkt  $(x_v, y_v)$  schneiden.

**Satz 4.1.** Die konvexe Kontur  $C_G$  einer Zeichnung  $\Gamma(G)$  eines Graphen  $G$  mit  $n$  Knoten kann in  $\mathcal{O}(k \cdot n)$  Zeit bestimmt werden, wenn für die Menge der Richtungen  $\mathcal{D}_k$  gilt  $k = |\mathcal{D}_k|$ .

*Beweis.* Für die Bestimmung einer  $\vec{D}$ -maximalen Gerade mit  $\vec{D} \in \mathcal{D}_k$  muss für jeden Knoten das Skalarprodukt berechnet und das Maximum daraus bestimmt werden. Somit ist dieser Schritt in  $\mathcal{O}(n)$  realisierbar. Die Bestimmung von  $F_G$  nimmt  $\mathcal{O}(k)$  Zeit, in Anspruch, so dass die Laufzeit zum Berechnen einer konvexen Kontur  $C_G$  einer Zeichnung eines Graphens  $\Gamma(G)$  in  $\mathcal{O}(k \cdot n)$  mit  $k = |\mathcal{D}_k|$  ist.  $\square$

In der Literatur findet sich bezüglich der Unterbaum Separation bisher die Rechteck Unterbaum Separation. Diese wurde für hierarchische Zeichnungen sowie für Zeichnungen auf dem 4-er Gitter definiert [CGKT02]. Für die Gitter mit sechs bzw. acht Richtungen ist es naheliegend diese Definition anzupassen. Deshalb wird für die  $k$ -Gitter im Sinne der Unterbaum Separation das kleinste umschließende  $k$ -Eck verwendet, das exakt der Definition 4.5 entspricht.

Auf dem 8-er Gitter ist das ein 8-eck, siehe Abbildung 4.6(b). Diese Art von Unterbaum Separation wird *k-Eck Unterbaum Separation* genannt. Ist ein Baum auf dem  $k$ -Gitter mit  $k$ -Eck Unterbaum Separation gezeichnet, so sind die kleinsten umschließenden konvexen  $k$ -Ecke der disjunkten Unterbäume ebenfalls disjunkt. Die Eigenschaft der  $k$ -Eck Unterbaum Separation wird im nachfolgend beschriebenen Verfahren erhalten, damit disjunkte Unterbäume in Zeichnungen leichter unterschieden werden können.

### 4.1.2 Beschreibung des Verfahrens

Im Folgenden wird Algorithmus 4.1 vorgestellt, der  $OLP_k$ -Zeichnungen von  $d$ -nären Bäumen auf dem  $k$ -Gitter für  $k \in \{4, 6, 8\}$  und  $d < k$  erstellt. Als Eingabe erwartet das Verfahren einen geordneten  $d$ -nären Baum  $T$  mit Wurzelknoten  $v$ . Die Ausgabe von Algorithmus 4.1 ist eine kompaktierte  $OLP_k$ -Zeichnung  $\Gamma(T)$ , die  $k$ -Eck Unterbaum Separation und isomorphe Unterbäume bis auf Verschiebung und Rotation identisch zeichnet.

---

#### Algorithmus 4.1. kompaktiereZeichnungMitSubtreeSeparation

---

**Input:** Wurzel  $v$  eines geordneten  $d$ -nären Baumes  $T = (V, E)$  mit  $k \in \{4, 6, 8\}$  und  $d < k$

**Output:** Kompaktierte Zeichnung  $\Gamma(T)$  von  $T$

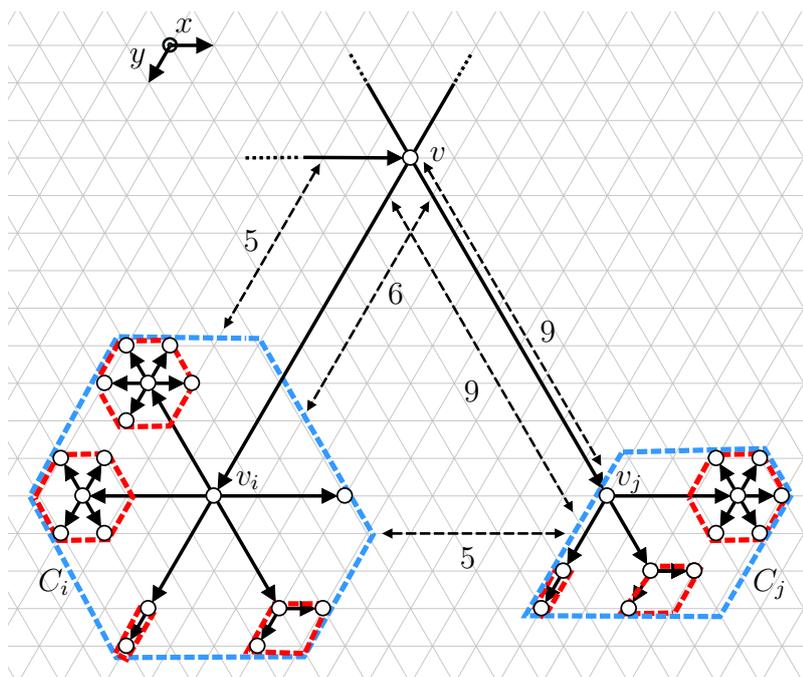
```

1 foreach  $w \in V$  do
2   |  $w$ .ausgehendeKantenLaenge  $\leftarrow 3^{h(T)-t(w)-1}$ ;
3 end
4 zeichneBaum( $T$ );
5 kompaktiereZeichnungMitSubtreeSeparation( $v$ );
6 zeichneBaum( $T$ );

```

---

Zuerst wird in Zeile 2 jeder ausgehenden Kante eines Knotens  $w \in V$  eine initiale Länge  $3^{h(T)-t(w)-1}$  zugewiesen, vgl. Beweis von Satz 3.8. Diese Kantenlängen stellen die Planarität der initialen Zeichnung sicher. In Zeile 4 wird initial gezeichnet, um dann in Zeile 5 die Zeichnung zu kompaktieren. In diesem Abschnitt



**Abbildung 4.7.** Abstandsberechnungen zweier Unterbäume  $T_{v_i}$  und  $T_{v_j}$  auf dem symmetrischen 6-er Gitter

wird die Kompaktierung mit dem Verfahren mit der  $k$ -Eck Unterbaum Separation *kompaktiereZeichnungMitSubtreeSeparation* durchgeführt. In der letzten Zeile 6 wird die kompaktierte Zeichnung in das entsprechende  $k$ -Gitter eingebettet.

Durch die Forderung des Ästhetikkriteriums der Pattern sind die Richtungen der ausgehenden Kanten eines Knotens  $v \in V$  relativ zur eingehenden Kante festgelegt. Somit können durch das Bestimmen der Richtung einer einzigen Kante alle absoluten Kantenrichtungen in einer Zeichnung bestimmt werden, was dann mit den in Zeile 2 berechneten Kantenlängen eine eindeutige Zeichnung liefert.

Der Algorithmus 4.2, der die Verkürzung der ausgehenden Kanten jedes Knotens berechnet, erhält die lokale Uniformität, weil die Länge der ausgehenden Kanten eines Knotens simultan verkürzt werden. Des Weiteren werden die von der initialen Zeichnung gegebenen Kantenrichtungen nicht verändert. Ziel ist es, unter Beachtung der  $k$ -Eck Unterbaum Separation die Kanten zu verkürzen, um die Fläche der Zeichnung zu reduzieren. Dies liefert dann eine lesbare und übersichtlichere Zeichnung des  $d$ -nären Baumes. Auf dem  $k$ -Gitter für  $k \in \{4, 6, 8\}$  werden Abstände entlang der Richtungen  $\mathcal{D}_k$  bestimmt, siehe Behauptung 4.1. Wie oben beschrieben muss beachtet werden, wie sich die Entfernung zwischen den Kindern von  $v$  bei einer Kantenverkürzung von eins verändert. Dieser Veränderungsfaktor kann aus der Länge  $|\vec{D}|$  abgelesen werden, siehe Abbildung 4.2.

Mit der Bestimmung der relativen Bewegungsrichtung  $\vec{D}$  und dem Verkürzungsfaktor  $|\vec{D}|$  kann die maximale Verkürzung ausgehender Kanten eines Knotens  $v$  be-

stimmt werden. Die ausgehenden Kanten werden so weit verkürzt, dass sich konvexe Konturen gerade nicht berühren bzw. überlagern und somit den Mindestabstand von eins einhalten. Dazu werden die Abstände zwischen Paaren von konvexen Konturen der Geschwisterknoten betrachtet. Die konvexen Konturen  $C_i$  und  $C_j$  der Geschwisterknoten  $v_i$  und  $v_j$  sind durch die Rekursion gegeben. Die Abstandsberechnung zwischen den konvexen Konturen  $C_i$  und  $C_j$  der Geschwister  $v_i$  und  $v_j$  ist durch den minimalen Abstand zwischen diesen beiden Konturen  $C_i$  und  $C_j$  entlang der Richtung  $\vec{D}_{i,j}$  festgelegt, siehe Abbildung 4.7. Da in diesem Fall  $|\vec{D}| = 1$ , können die ausgehenden Kanten um den Abstand  $d(C_i, C_j)$  minus 1, also  $5 - 1 = 4$  verkürzt werden, ohne eine Überlagerung der konvexen Konturen  $C_i$  und  $C_j$  zu erzeugen.

Des Weiteren müssen noch Abstände zur ggf. vorhandenen eingehenden Kante von  $v$  und den Geschwisterkanten berücksichtigt werden. Dabei wird in der Distanzberechnung immer die relative Bewegungsrichtung verwendet. Die eingehende Kante von  $v$  ändert die Länge oder Lage bei einer Verkürzung der ausgehenden Kanten von  $v$  nicht, so dass die Bewegungsrichtung einer Kontur  $C_i$  relativ zu  $v$   $\vec{D} = -\vec{v}_i$  ist. In Abbildung 4.7 sind die Bewegungsrichtungen gestrichelt eingezeichnet.

Im Folgenden wird Algorithmus 4.2 detailliert beschrieben, der rekursiv die Verkürzung der Kantenlängen für einen Baum  $T$  mit der Wurzel  $v$  berechnet. Zunächst wird in Zeile 1 die triviale Kontur  $C$  für den Knoten  $v$  und in Zeile 2 wird eine leere Menge  $\mathcal{C}$  zum späteren Zwischenspeichern der Konturen der Kindunterbäume  $T_{v_1}, \dots, T_{v_i}$  von  $v$  mit  $i \leq d$  erstellt.

Der rekursive Aufruf von *kompaktiereZeichnungMitUnterbaumSeparation* erfolgt in den Zeilen 3 bis 7, um die Verkürzung der Kanten der Kindunterbäume und deren konvexen Konturen zu berechnen. Der Verkürzungswert *trim* wird in Zeile 8 mit  $\infty$  initialisiert. In den Zeilen 9 bis 13 werden mit Hilfe der Abstände zwischen allen Paaren von Konturen der Kinder  $C_i$  und  $C_j$  die maximal möglichen Verkürzungswerte bestimmt. Dabei wird zuerst in Zeile 10 der Abstand zwischen der Kontur  $C_i$  von  $T_{v_i}$  und der eingehenden Kante von  $v_j$  bestimmt und dann *trim* aktualisiert. Geschwisterknoten  $v_i$  und  $v_j$  bewegen sich bei einer simulatanen Verkürzung der eingehenden Kanten von  $v_i$  und  $v_j$  in Richtung  $\vec{D}_{i,j}$  aufeinander zu, siehe Zeile 11. Der Wert  $|\vec{D}_{i,j}|$  gibt an, um wieviel sich zwei Unterbäume  $T_{v_i}$  und  $T_{v_j}$  annähern, wenn die eingehenden Kanten von  $v_i$  und  $v_j$  um eins verkürzt werden. In Zeile 12 wird *trim* aktualisiert, falls der Abstand zwischen  $C_i$  und  $C_j$  in Richtung  $\vec{D}_{i,j}$  es erfordert. Wenn  $v$  einen Vaterknoten besitzt, dann wird der Abstand zur eingehenden Kante von  $v$  mit berücksichtigt, so dass bei der Verkürzung die Kontur  $C_i$  diese nicht schneidet (Zeilen 14 bis 18). Ansonsten wird verhindert, dass die Kontur sich nicht mit  $v$  überlagert (Zeilen 19 bis 23).

Der Wert *trim* gibt jetzt an, um wieviel die ausgehenden Kanten von  $v$  verkürzt werden können, ohne dabei die Planarität zu zerstören und ohne dass sich Paare konvexer Konturen von Geschwisterunterbäumen überlappen. In den Zeilen 24 bis 27 werden die Konturen der Kinder um den Wert *trim* in Richtung  $v$  verschoben. Es ist zu beachten, dass nicht alle Knoten der Unterbäume verschoben werden, da dies die Laufzeit entscheidend verschlechtern würde. Zum Erstellen der konvexen

---

**Algorithmus 4.2.** kompaktiereZeichnungMitUnterbaumSeparation

---

**Input:** Wurzelknoten  $v$  eines geordneten  $d$ -nären Baumes mit  $k \in \{4, 6, 8\}$   
und  $d < k$

**Output:** Konvexe Kontur  $C_v$  des Unterbaums  $T_v$  und die verkürzten  
Kantenlängen von  $T_v$

```

1 KonvexeKontur  $C_v \leftarrow \text{KonvexeKontur}(v)$ ;
2 Set  $\mathcal{C} \leftarrow \emptyset$ ;
3 foreach Kind  $v_i$  von  $v$  do
4   | // rekursiver Aufruf
5   |  $C_i \leftarrow \text{kompaktiereZeichnungMitUnterbaumSeparation}(v_i)$ ;
6   |  $\mathcal{C} = \mathcal{C} \cup \{C_i\}$ ;
7 end
8  $\text{trim} \leftarrow \infty$ ;
9 foreach  $C_i, C_j (i \neq j) \in \mathcal{C}$  do
10  |  $\text{trim} \leftarrow \min(\text{trim}, d_{(v,v_i)}((v, v_j), C_i) - 1)$ ;
11  | Vektor  $\vec{D} \leftarrow \frac{\vec{v}_i}{|v_i|} - \frac{\vec{v}_j}{|v_j|}$ ; // relative Bewegungsrichtung
12  |  $\text{trim} \leftarrow \min\left(\text{trim}, \left\lfloor \frac{d_{\vec{D}}(C_i, C_j) - 1}{|\vec{D}|} \right\rfloor\right)$ ; // Update der Variable trim
13 end
14 if  $v.Vater \neq \text{nil}$  then
15  | foreach  $C_i \in \mathcal{C}$  do
16  | |  $\text{trim} \leftarrow \min(\text{trim}, d_{(v,v_i)}((v, v.Vater), C_i) - 1)$ ;
17  | end
18 end
19 else
20  | foreach  $C_i \in \mathcal{C}$  do
21  | |  $\text{trim} \leftarrow \min(\text{trim}, d_{(v,v_i)}(v, C_i) - 1)$ ;
22  | end
23 end
24 foreach  $C_s \in \mathcal{C}$  do
25  |  $\text{verschiebe}(C_s, \text{trim})$ ;
26  |  $C_v.\text{mergeKontur}(C_s)$ ;
27 end
28  $v.\text{ausgehendeKantenlaenge} \leftarrow v.\text{ausgehendeKantenlaenge} - \text{trim}$ ;
29 return  $C_v$ ;

```

---

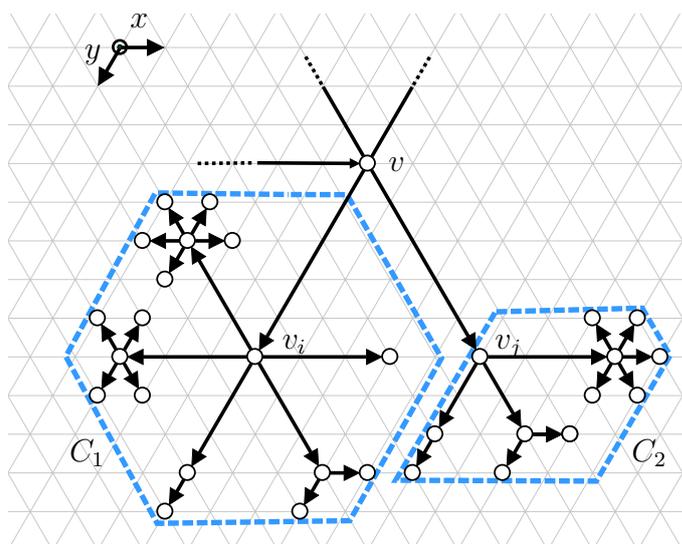
Kontur  $C_v$  des Unterbaumes  $T_v$  werden die bereits rekursiv berechneten Konturen der Kinderunterbäume aus  $\mathcal{C}$  verwendet. Dazu werden für jede Richtung  $\vec{D} \in \mathcal{D}_k$  die  $\vec{D}$ -maximalen Knoten bzw. die parallelen  $\vec{D}$ -maximalen Geraden verwendet, um aus diesen wieder die  $\vec{D}$ -maximale Gerade mit Blick auf den Unterbaum  $T_v$  zu bestimmen. Das sind für  $\vec{D}$  höchstens  $k$  Elemente ( $i \leq d < k$  Kinder von  $v$  und die triviale Kontur von  $v$ ), so dass dies in  $\mathcal{O}(k)$  Zeit bestimmt wird, siehe Zeile 26. Zuletzt wird in Zeile 28 die Länge der ausgehenden Kanten von  $v$  um den Wert *trim* verkürzt und in Zeile 29 die aktualisierte Kontur  $C$  zurückgegeben. Die Berechnung eines Rekursionsschrittes wird mit Hilfe von Abbildung 4.7 erläutert, wobei angenommen wird, dass die nicht eingezeichneten Kinder von  $v$  die Berechnungen nicht beeinflussen. Darin sind folgende Abstände gegeben:

1. Aus den Abständen der Konturen zu den Geschwisterkanten (siehe Zeile 10) ergibt sich ein maximaler Verkürzungswert von fünf:
  - Abstand der konvexen Kontur  $C_i$  zur Kante  $(v, v_j)$  in Richtung  $(v, v_i)$   
 $d_{\vec{v}_i}((v, v_j), C_i) = 6$
  - Abstand der konvexen Kontur  $C_j$  zur Kante  $(v, v_i)$  in Richtung  $(v, v_j)$   
 $d_{\vec{v}_j}((v, v_i), C_j) = 9$
2. Aus dem Abstand zwischen den konvexen Konturen  $C_1$  und  $C_2$  mit  $\vec{D} = \vec{v}_i - \vec{v}_j$   
 $d_{\vec{D}}(C_i, C_j) = 5$  und  $|\vec{D}| = 1$  ergibt sich ein maximaler Verkürzungswert von vier (Zeilen 11 und 12)
3. Der Abstand der Konturen  $C_i$  und  $C_j$  zur eingehenden Kante von  $v$  liefert einen maximalen Verkürzungswert von vier (Zeile 16)
  - $d_{\vec{v}_i}((v, v.Vater), C_i) = 5$
  - $d_{\vec{v}_j}((v, v.Vater), C_j) = 9$

Das Minimum der maximalen Verkürzungswerte ist vier. Somit werden die ausgehenden Kanten von  $v$  um den Wert *trim* = 4 verkürzt (Zeile 28). Das Resultat ist in Abbildung 4.8 gezeigt. Würde man die ausgehenden Kanten von  $v$  noch weiter verkürzen, so ergäbe sich eine Überlagerung der konvexen Konturen  $C_1$  und  $C_2$ . Dies hätte zur Folge, dass die Eigenschaft der  $k$ -Eck Unterbaum Separation verletzen würde. Durch die Verwendung der konvexen Konturen wird die Planarität einer Zeichnung sichergestellt, da sich Knoten einer Zeichnung nur innerhalb des konvexen  $k$ -Ecks befinden und im Verfahren verhindert wird, dass sich  $k$ -Ecke bzw. konvexe Konturen von Geschwisterknoten überlagern. Des Weiteren werden identische Unterbäume durch das rekursive Vorgehen identisch bis auf Verschiebung oder Rotation gezeichnet.

### 4.1.3 Zeitkomplexität

Das beschriebene Verfahren, das konvexe Konturen zur Erfüllung des Ästhetikkriteriums der  $k$ -Eck Unterbaum Separation verwendet, kann effizient in Linearzeit



**Abbildung 4.8.** Verkürzung der Kanten  $(v, v_i)$  und  $(v, v_j)$  aus Abbildung 4.7

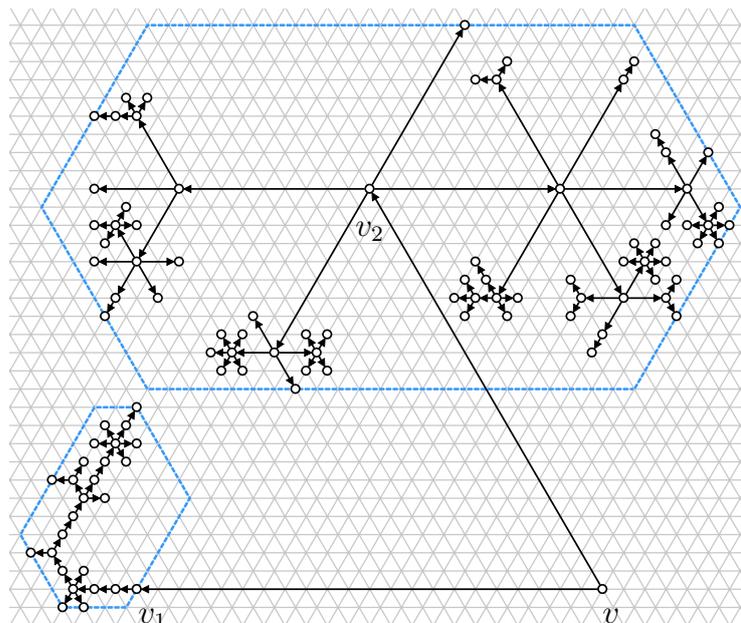
implementiert werden, da die konvexe Kontur eines Unterbaumes lediglich aus  $2k$  verschiedenen Koordinaten besteht.

**Satz 4.2.** Sei  $T$  ein beliebiger  $d$ -närer Baum mit  $n$  Knoten,  $k \in \{4, 6, 8\}$  und  $d < k$  sowie der Wurzel  $v$ . Algorithmus 4.1 mit dem Aufruf des rekursiven Algorithmus 4.2 hat eine Zeitkomplexität in  $\mathcal{O}(n)$ .

*Beweis.* Die Laufzeit jeder Zeile im Algorithmus 4.1 ist in  $\mathcal{O}(n)$ . Das Zuweisen der initialen Kantenlängen sowie das Zeichnen auf dem Gitter erfordert ein Traversieren des Baumes. Jeder Rekursionsschritt des Algorithmus führt für die Unterbäume der Kinder von  $v$  Distanzberechnungen aus, die  $\mathcal{O}(1)$  Zeit benötigen, weil die Anzahl der Segmente einer Kontur in  $\mathcal{O}(1)$  ist.

Jeder Knoten hat höchstens  $d < k$  Kinder und jede konvexe Kontur besteht aus höchstens  $k$  Segmenten. Da beim Verkürzen der Kanten nicht der komplette Unterbaum verschoben wird, ist dies durch das Verschieben des  $k$ -Ecks ebenfalls in konstanter Zeit möglich. Somit wird jeder einzelne Rekursionsschritt in konstanter Zeit ausgeführt. Die rekursive Methode von Algorithmus 4.2 wird für jeden Knoten einmal aufgerufen, was für einen Baum  $T$  mit  $n$  Knoten zu einer Gesamtlaufzeit in  $\mathcal{O}(n)$  führt.  $\square$

Mit dieser Heuristik können für das 4-er Gitter, das 6-er Gitter und das 8-er Gitter Zeichnungen in Linearzeit generiert werden. Dies dient dazu, dass die Fläche verringert wird und dabei die Lesbarkeit und Übersichtlichkeit verbessert wird. Gleichzeitig bleiben die Eigenschaft der  $k$ -Eck Unterbaum Separation sowie die genannten Ästhetikkriterien erhalten.



**Abbildung 4.9.** Negativbeispiel von Algorithmus 4.2 auf dem symmetrischen 6-er Gitter

#### 4.1.4 Mögliche Erweiterungen

Der Algorithmus 4.2 kann ebenfalls auf bereits bestehende Zeichnungen angewendet werden, deren Fläche verringert werden soll. Die Voraussetzung dazu ist, dass die  $k$ -Eck Unterbaum Separation in der Eingabezeichnung erfüllt ist. Es ist nicht notwendig, dass die Zeichnungen ordnungserhaltend, lokal uniform und/oder nach Pattern gezeichnet sind. Ist eines oder mehrere dieser Kriterien in der Eingabezeichnung erfüllt, so wird die Resultatzeichnung von Algorithmus 4.2 diese Eigenschaften erhalten.

Der Vorteil der Unterbaum Separation hat jedoch auch einen Nachteil. Dieser besteht darin, dass die konvexen Konturen zwar effizient berechnet werden können und die  $k$ -Eck Unterbaum Separation erhalten bleibt, aber die Konturen nicht exakt genug sind, um im Rekursionsschritt die ausgehenden Kanten maximal zu verkürzen, siehe dazu Abbildung 4.9. Die eingezeichneten konvexen Konturen verhindern eine weitere Verkürzung der ausgehenden Kanten von  $v$ . Die ausgehenden Kanten von  $v$  können jedoch weiter verkürzt werden. Dies ist oft wünschenswert, um Zeichnungen mit kleinerer Fläche zu erzeugen. Dabei werden die Planarität und die restlichen Ästhetikkriterien, bis auf die  $k$ -Eck Unterbaum Separation beachtet, siehe Abbildung 4.9. Ohne Beachtung der  $k$ -Eck Unterbaum Separation ist es notwendig, eine geeignetere Datenstruktur für die Abstandsberechnung zu verwenden. Diese Problemstellung wird im nächsten Abschnitt 4.2 behandelt und ein geeignetes Verfahren dazu vorgestellt.

## 4.2 Zeichnungen von $d$ -nären Bäumen ohne Unterbaum Separation

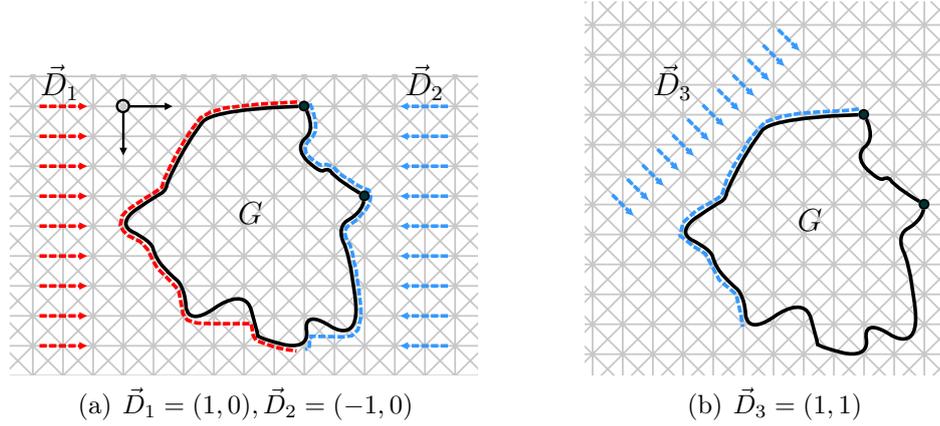
In diesem Abschnitt wird ebenfalls eine Heuristik für das Zeichnen  $d$ -närer Bäume auf dem  $k$ -Gitter mit  $k \in \{4, 6, 8\}$  und  $d < k$  präsentiert. Dies stellt eine Verbesserung zum Algorithmus 4.1 aus Abschnitt 4.1 hinsichtlich der benötigten Fläche dar. Den Unterschied zur Heuristik Algorithmus 4.2 zeigt das Beispiel in Abbildung 4.9, wo zwei Unterbäume und deren konvexe Konturen dargestellt sind. Nach Ausführung von Algorithmus 4.2 können die ausgehenden Kanten  $(v, v_i)$  und  $(v, v_j)$  aufgrund der  $k$ -Eck Unterbaum Separation nicht weiter verkürzt werden, da der Abstand der konvexen Konturen  $C_i$  und  $C_j$  bereits minimal ist. In Abbildung 4.9 ist zu sehen, dass  $(v, v_i)$  und  $(v, v_j)$  jedoch noch weiter verkürzt werden können, ohne die Planarität zu zerstören.

Im Folgenden wird ein Verfahren vorgestellt, das die ausgehenden Kanten von Knoten so weit verkürzt, dass die Planarität gerade noch erhalten bleibt und als Ergebnis eine  $OLP_k$ -Zeichnung mit den Ästhetikkriterien ganzzahlige Koordinaten, Planarität sowie Mindestabstand zwischen Knoten produziert. Des Weiteren werden isomorphe Unterbäume bis auf Verschiebung und Rotation identisch gezeichnet. Ein Beispiel zeigt Abbildung 4.12. Dazu wird die Zeile 5 in Algorithmus 4.1 mit dem rekursiven Aufruf *kompaktiereZeichnungOhneSubtreeSeparation*( $v$ ) von Algorithmus 4.3 ersetzt.

Algorithmus 4.3 verwendet zur Berechnung keine konvexen Konturen, sondern exaktere Konturen, die im Folgenden definiert werden (Definition 4.6). Dadurch ist es im Rekursionsschritt möglich, die ausgehenden Kanten eines Knotens ggf. weiter zu verkürzen, ohne die Planarität zu zerstören. Dies führt in der Regel zu einer kleineren Fläche der resultierenden Zeichnung. Der entscheidende Unterschied ist die Anwendung von Konturen, wie sie in ähnlicher Weise bereits im hierarchischen Fall verwendet werden [RT81, Wal90, BJL02]. Im hierarchischen Fall sind die Konturen einer Zeichnung eines Unterbaumes die Liste der linkesten Knoten und die Liste der rechtesten Knoten der Level. Im Gegensatz zu den Konturen in hierarchischen Baumzeichnungen, ist es jedoch notwendig, für jede der verschiedenen relativen Bewegungsrichtungen, Richtungskonturen für die Abstandsberechnung zu definieren.

### 4.2.1 Definitionen

Zur Definition einer weiteren Datenstruktur wird der Ansatz aus der Sichtbarkeits- und Bewegungsplanung verwendet [Her89]. Gegeben sind undurchsichtige Hindernisse auf bestimmten Koordinaten in der Ebene. Gesucht wird der *obere Rand*  $S$  davon, der aus einer bestimmten Richtung, z. B. vom Punkt  $P = (0, +\infty)$ , also von oben, sichtbar ist. Die Menge  $S$  von sich nicht kreuzenden Liniensegmenten, die von  $P$  aus sichtbaren Teile, können in  $\mathcal{O}(n \log n)$  Zeit berechnet werden. Ist die Reihenfolge in  $x$ -Richtung gegeben, so kann dies auch in  $\mathcal{O}(n)$  berechnet werden. Kreuzen sich die Segmente, so kann der sichtbare Teil in  $\mathcal{O}(n\alpha(n))$  berechnet werden, wobei



**Abbildung 4.10.** Richtungskonturen für die  $D_1, D_2, D_3 \in \mathcal{D}_8$

$\alpha(n)$  die inverse Funktion zur Ackermannfunktion ist. Im Folgenden wird der obere Rand für verschiedene Richtungen  $D \in \mathcal{D}_k$  in Definition 4.6 erweitert und als *Kontur* bezeichnet.

**Definition 4.6 (Richtungskontur, Kontur).** Eine Richtungskontur  $C_{G, \vec{D}}$  einer Zeichnung eines Graphens  $G = (V, E)$  mit  $\vec{D} \in \mathcal{D}_k$  ist eine monoton steigende Polylinie in Bezug auf  $\vec{D}^\circ$ .  $C_{G, \vec{D}}$  ist die Menge von Knoten und Kantensegmenten der Zeichnung  $\Gamma(G)$ , die aus einer bestimmten Richtung  $\vec{D} \in \mathcal{D}_k$  sichtbar ist. Die Kontur  $C_G = \{C_{G, \vec{D}} \mid \vec{D} \in \mathcal{D}_k\}$  eines zusammenhängenden Graphens  $G = (V, E)$  ist die Menge der Richtungskonturen für  $\vec{D} \in \mathcal{D}_k$ .

Im Folgenden werden in Zeichnungen Kantensegmente oder Knoten auch als *Segmente* oder *Punkte* bezeichnet. Wenn zwei benachbarte Punkte oder Segmente einer Richtungskontur nicht denselben Endpunkt haben, werden sie durch ein Segment parallel zu  $\vec{D}$  verbunden. Die Richtungskontur eines einzelnen Knotens  $v$  in einer Zeichnung besteht für jede Richtung  $\vec{D} \in \mathcal{D}_k$  aus seinen Koordinaten  $(x_v, y_v)$ .

In Abbildung 4.11 sind die sichtbaren Segmente  $S_1, S_3, S_4, S_6, S_7, S_9, S_{10}, S_{11}$  aus der Richtung  $\vec{D} = (-1, 1)$  in einer Zeichnung auf dem 8-er Gitter gezeigt. Darin verbindet das zu  $\vec{D}$  parallele Segment  $S_2$  die benachbarten Segmente  $S_1$  und  $S_3$  (analog die zu  $\vec{D}$  parallelen Segmente  $S_5$  und  $S_8$ ), so dass daraus die Kontur  $C_{\vec{D}}$  resultiert.

**Lemma 4.1.** Die Anzahl der Segmente in einer Richtungskontur  $C_{v, \vec{D}}$  einer Zeichnung  $\Gamma(T_v)$  eines Unterbaumes  $T_v$  mit  $\vec{D} \in \mathcal{D}_k$  ist linear in der Anzahl der Knoten von  $T_v$ .

*Beweis.* Betrachte die Kontur  $C_{v, \vec{D}}$  der Richtung  $\vec{D}$  einer Zeichnung  $\Gamma(T_v)$  eines Baumes  $T_v$ . Aufgrund der Geradlinigkeit der Kanten und des Zusammenhangs von  $T_v$  erzeugt jede Kante und jeder Knoten höchstens ein dazugehöriges Segment.  $\square$

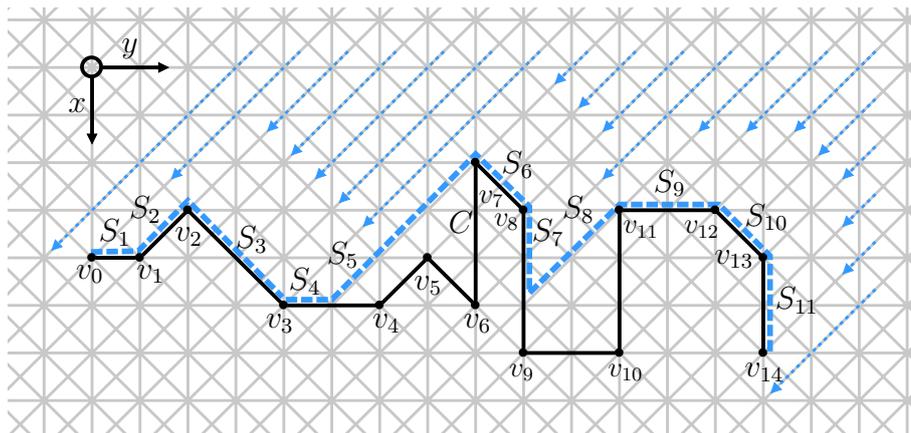


Abbildung 4.11. Kontur  $C_{\vec{D}}$  eines Pfades für die Richtung  $\vec{D} = (-1, 1)$

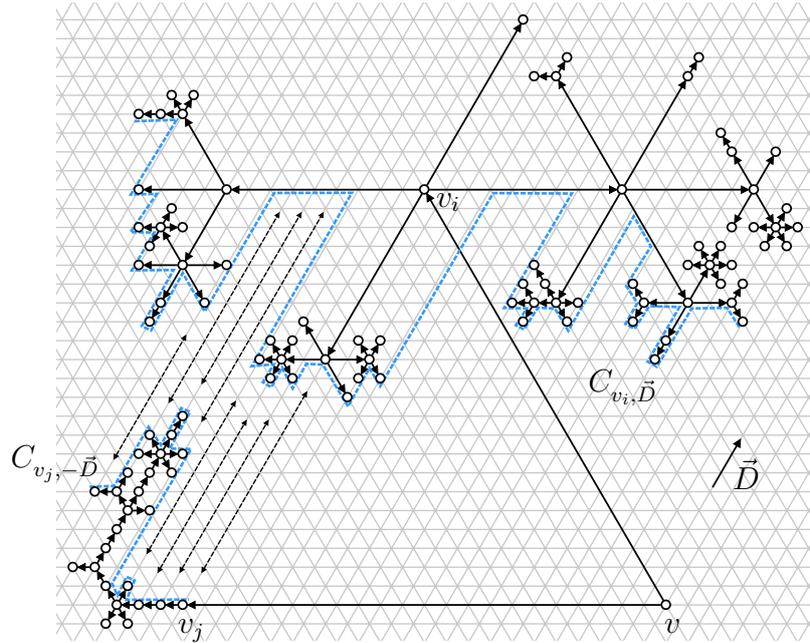
Im folgenden Abschnitt werden Richtungskonturen verwendet, um die Distanzberechnung zwischen Unterbäumen zu berechnen. Diese Richtungskonturen werden als verkettete Listen gespeichert.

### 4.2.2 Beschreibung des Verfahrens

Wie in Abschnitt 4.1 werden  $d$ -näre Bäume in einem rekursiven Zeichenverfahren auf dem  $k$ -Gitter mit  $k \in \{4, 6, 8\}$  und  $d < k$  eingebettet, mit dem entscheidenden Unterschied, dass zur Distanzberechnung zwischen Unterbäumen Richtungskonturen verwendet werden. Es wird ein maximal möglicher Verkürzungswert mit den Richtungskonturen bestimmt. Die ausgehenden Kanten eines Knotens  $v$  werden wieder simultan verkürzt, so dass die Bewegungsrichtung zweier Kinder  $v_i$  und  $v_j$  relativ zueinander festgelegt ist durch  $\vec{D} = \frac{\vec{v}_i}{|\vec{v}_i|} - \frac{\vec{v}_j}{|\vec{v}_j|}$ , wie in Abschnitt 4.1. Ein Beispiel ist in Abbildung 4.12 gezeigt, wo die Richtungsabstände entlang  $\vec{D}$  zwischen den Mengen von Segmenten der Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, -\vec{D}}$  eingezeichnet sind. Der minimale Wert der Richtungsabstände  $dist = \min\{d_{\vec{D}}(S_i, S_j) \mid S_i \in C_{v_i, \vec{D}}, S_j \in C_{v_j, -\vec{D}}\}$  wird zwischen allen Paaren von Segmenten  $S_i \in C_{v_i, \vec{D}}$  und  $S_j \in C_{v_j, -\vec{D}}$  berechnet.

Abhängig von  $|\vec{D}|$  werden die ausgehenden Kanten um  $\lfloor \frac{dist-1}{|\vec{D}|} \rfloor$  verkürzt. Durch die Abstandsberechnung ist sichergestellt, dass die Planarität erhalten bleibt. Der Abstand zweier Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, -\vec{D}}$  kann in  $\mathcal{O}(n)$  Zeit unter Verwendung eines Scanline-Verfahrens berechnet werden, indem beide Listen  $C_{v_i, \vec{D}}$  und  $C_{v_j, -\vec{D}}$  entlang der zu  $\vec{D}$  orthogonalen Richtung  $\vec{D}^\circ$  abgelaufen werden. Dies ist möglich, da die Richtungskonturen entlang  $\vec{D}^\circ$  nach Definition monoton sind.

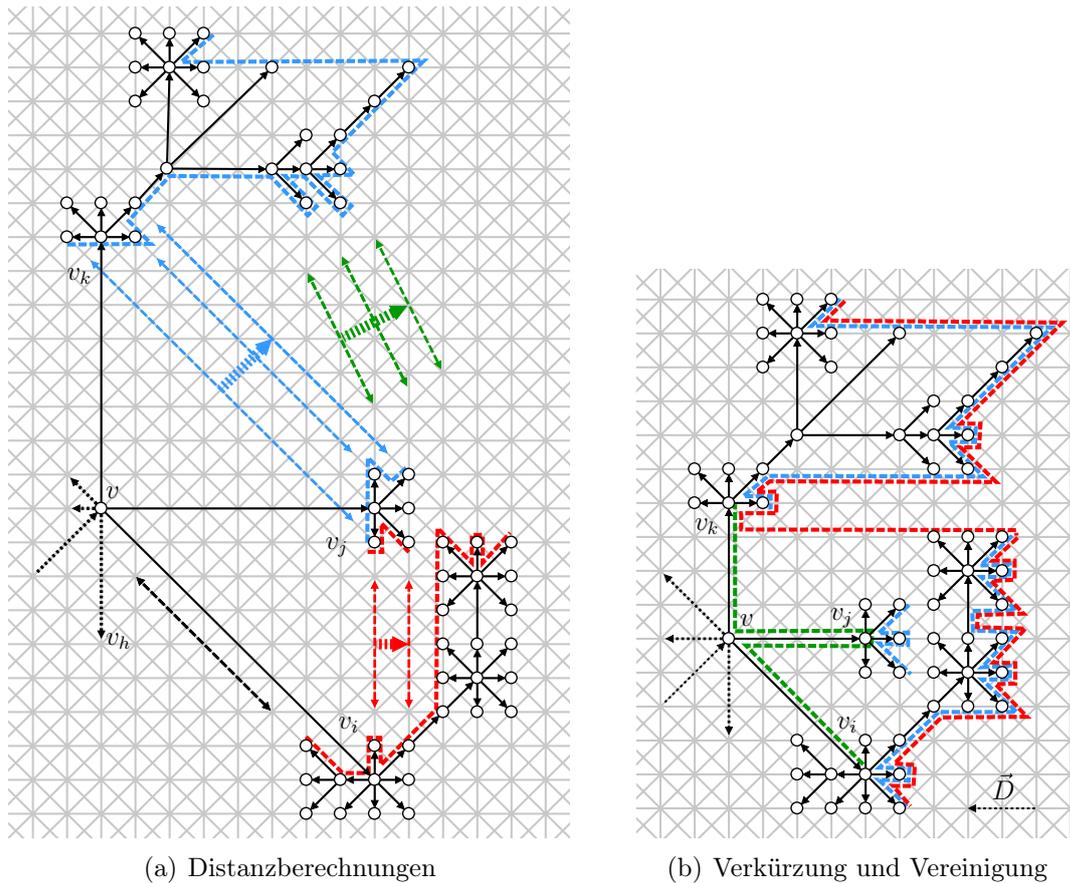
Im rekursiven Algorithmus 4.3 wird für ein Blatt  $v$  die triviale Kontur  $C_v$  (Zeile 1) initialisiert und in Zeile 2 wird die Menge  $\mathcal{C}$  der potentiellen Konturen der Unterbäume der Kinder initialisiert. Der rekursive Aufruf für jedes Kind  $v_i$  der inneren



**Abbildung 4.12.** Distanzberechnung mit exakten Konturen zwischen  $C_{v_i, \vec{D}}$  und  $C_{v_j, -\vec{D}}$  auf dem 6-er Gitter

Knoten erfolgt in den Zeilen 3 bis 7.

Die maximal mögliche Verkürzung der ausgehenden Kanten des Knotens  $v$  bestimmt am Ende der Wert von  $trim$ , der in Zeile 8 initialisiert wird. In den Zeilen 9 bis 16 werden die Distanzen zwischen allen Paaren von Konturen mit den notwendigen Richtungen bestimmt. Dazu wird zuerst der Abstand der Richtungskontur  $C_{i, \vec{v}_i}$  und der Kante  $(v, v_j)$  bestimmt. In Zeile 12 wird im Richtungsvektor  $\vec{D}$  gespeichert, entlang welcher Richtung sich, relativ von  $v_i$  aus gesehen, die beiden Unterbäume bei einer simultanen Verkürzung zueinander bewegen. Dabei gibt  $|\vec{D}|$ , wie auch schon in Abschnitt 4.1, an, um welchen Wert sich das Paar von Unterbäumen bei einer Verkürzung um eins aufeinander zu bewegen, was im Folgenden dazu verwendet wird, um die maximalen Verkürzungswert zu bestimmen, siehe Zeile 13. In Zeile 14 ist der Unterschied zu Algorithmus 4.2, dass jetzt nicht mehr die konvexen Konturen zur Berechnung des Abstandes verwendet werden, sondern nur die Richtungskonturen der Richtungen  $\vec{D}$  und  $-\vec{D}$  der Geschwisterunterbäume  $T_{v_i}$  und  $T_{v_j}$ , die einander „zugewandt“ sind, siehe dazu  $C_{v_i, \vec{D}}$  und  $C_{v_j, -\vec{D}}$  in Abbildung 4.12. Konkret werden diese Richtungen durch den Vektor  $\vec{D}$  bestimmt, da sich die beiden Unterbäume bei einer simultanen Kantenverkürzung in dieser Richtung aufeinander zu bewegen. Zeile 15 aktualisiert den Wert von  $trim$ , wie es auch in Algorithmus 4.2 gemacht wird. Falls  $v$  einen Vaterknoten besitzt, so wird der Abstand zwischen der Kante  $(v.Vater, v)$  berücksichtigt, um eine Überlagerung der Konturen mit dieser zu vermeiden (Zeilen 18 bis 22). Andernfalls wird der Abstand zum Knoten  $v$  berechnet



**Abbildung 4.13.** Beispiel zu den Berechnungen im Rekursionsschritt mit Kindunterbäumen  $T(v_i)$ ,  $T(v_j)$  und  $T(v_k)$  von  $v$

und der Wert von *trim* aktualisiert (siehe Zeilen 24 bis 28). In den Zeilen 29 bis 32 wird jede Richtungskontur  $C_{v_i, \vec{D}}$  mit  $\vec{D} \in \mathcal{D}_k$  und  $k \in \{4, 6, 8\}$  um *trim* in Richtung  $-\vec{v}_i$  verschoben. Zuletzt wird in Zeile 33 die ausgehende Kante von  $v$  um *trim* verkürzt und die aktuelle Kontur  $C_v$  des Unterbaumes  $T_v$  zurückgegeben.

Die in diesem Abschnitt verwendete Kontur hat zur Folge, dass sich der Ablauf der Distanzberechnung und somit das Verkürzen der Kanten ändert. Des Weiteren ist es notwendig, das Zusammenführen der Konturen der Kindunterbäume zur Kontur  $C_v$  des aktuellen Knotens  $v$  im Rekursionsschritt anders zu gestalten. Beides wird in den folgenden Abschnitten 4.2.2.1 und 4.2.2.2 detaillierter ausgeführt.

#### 4.2.2.1 Verkürzen der Kanten

Um den maximal möglichen Verkürzungswert der ausgehenden Kanten eines Knotens  $v$  in einer Zeichnung  $\Gamma(T_v)$  zu bestimmen, werden die Abstände zwischen allen Paaren von Unterbäumen, also deren Konturen, und die Abstände jedes Unterbaumes zu den eingehenden Kanten der Geschwisterunterbäume benötigt. Seien  $v_i$  und

---

**Algorithmus 4.3.** kompaktiereZeichnungOhneUnterbaumSeparation
 

---

**Input:** Wurzelknoten  $v$  eines geordneten  $d$ -nären Baumes mit  $k \in \{4, 6, 8\}$   
und  $d < k$

**Output:** Kontur  $C_v$  des Knotens  $v$  und die verkürzten Kantenlängen von  $T_v$

```

1 Kontur  $C_v \leftarrow \text{Kontur}(v)$ ;
2 Set  $\mathcal{C} \leftarrow \emptyset$ ;
3 foreach Kind  $v_i$  von  $v$  do
4   | // rekursiver Aufruf
5   |  $C_i \leftarrow \text{kompaktiereZeichnungOhneUnterbaumSeparation}(v_i)$ ;
6   |  $\mathcal{C} = \mathcal{C} \cup \{C_i\}$ ;
7 end
8  $\text{trim} \leftarrow \infty$ ;
9 foreach  $C_i, C_j (i \neq j) \in \mathcal{C}$  do
10  | // Abstand zwischen Kante  $(v, v_j)$  und Richtungskontur  $C_{i, \vec{r}_i}$ 
11  |  $\text{trim} \leftarrow \min(\text{trim}, d_{\vec{v}_i}((v, v_j), C_{i, -\vec{v}_i}) - 1)$ ;
12  | Vektor  $\vec{D} \leftarrow \frac{\vec{v}_i}{|\vec{v}_i|} - \frac{\vec{v}_j}{|\vec{v}_j|}$ ; // relative Bewegungsrichtung
13  | Integer  $\text{laenge} \leftarrow |\vec{D}|$ ; // Länge
14  | Integer  $\text{dist} \leftarrow d_{\vec{D}}(C_{i, \vec{D}}, C_{j, -\vec{D}})$ ;
15  |  $\text{trim} \leftarrow \min\left(\text{trim}, \left\lfloor \frac{\text{dist}-1}{\text{laenge}} \right\rfloor\right)$ ; // Maximale Verkürzung
16 end
17 // Vaterknoten vorhanden
18 if  $v.Vater \neq \text{nil}$  then
19  | foreach  $C_i \in \mathcal{C}$  do
20  | |  $\text{trim} \leftarrow \min(\text{trim}, d_{\vec{v}_i}((v, v.Vater), C_{i, \vec{v}_i}) - 1)$ ;
21  | end
22 end
23 // kein Vaterknoten
24 else
25  | foreach  $C_i \in \mathcal{C}$  do
26  | |  $\text{trim} \leftarrow \min(\text{trim}, d_{\vec{v}_i}(v, C_{i, \vec{v}_i}) - 1)$ ;
27  | end
28 end
29 foreach  $C_s \in \mathcal{C}$  do
30  |  $\text{verschiebe}(C_s, \text{trim})$ ;
31  |  $C_v.\text{mergeKontur}(C_s)$ ;
32 end
33  $v.\text{ausgehendeKantenlaenge} \leftarrow v.\text{ausgehendeKantenlaenge} - \text{trim}$ ;
34 return  $C_v$ ;

```

---

$v_j$  Kinder von  $v$  mit den entsprechenden eingehenden Kanten  $e_i$  und  $e_j$ . Zuerst wird die minimale Distanz zwischen den Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, -\vec{D}}$  der Unterbäume  $T_{v_i}$  und  $T_{v_j}$  mit  $\vec{D}_{i,j} = \frac{\vec{v}_i}{|\vec{v}_i|} - \frac{\vec{v}_j}{|\vec{v}_j|}$  bestimmt. Der Richtungsvektor  $\vec{D}_{i,j}$  legt fest, welche Richtungskonturen sich gegenüber liegen. Dabei wird die Minimaldistanz zwischen  $C_{v_i, \vec{D}_{i,j}}$  und  $C_{v_j, -\vec{D}_{i,j}}$  mit einem Scanline Verfahren entlang  $\vec{D}_{i,j}^o$  berechnet. Jedes Segment der beiden Konturen wird einmal besucht, um den Abstand zur anderen Richtungskontur entlang  $\vec{D}_{i,j}$  zu bestimmen. Die Zeitkomplexität dieses Schrittes ist in der Anzahl der Knicke der Richtungskonturen und ist somit nach Lemma 4.1 linear in der Anzahl der Knoten. Der maximal mögliche Verkürzungswert zweier Geschwisterunterbäume  $T_{v_i}$  und  $T_{v_j}$  ist abhängig von der Länge von  $|\vec{D}_{i,j}|$  von  $\vec{D}_{i,j} = \frac{\vec{v}_i}{|\vec{v}_i|} - \frac{\vec{v}_j}{|\vec{v}_j|}$ , siehe Definition 4.1.

In Abbildung 4.13(a) können die Kanten  $(v, v_i)$  und  $(v, v_k)$  um  $\left\lfloor \frac{(\minDist-1)}{|\vec{D}_{i,j}|} \right\rfloor$  verkürzt werden, wobei  $\minDist$  der kleinste Abstand zwischen den entsprechenden Konturen und  $|\vec{D}_{i,j}| = 2$  ist. Ein Beispiel für  $|\vec{D}| = 1$  sind die Unterbäume  $T_i$  und  $T_j$  bzw.  $T_j$  und  $T_k$ . Der maximal mögliche Verkürzungswert wird für alle Paare von Unterbäumen der Kinder von  $v$  berechnet. Beispielsweise ist der minimale Abstand zwischen  $C_{v_j, -\vec{D}_{j,k}}$  und  $C_{v_k, \vec{D}_{j,k}}$  gleich sieben, so dass bei einer Verkürzung der ausgehenden Kanten von  $v$  die Planarität erhalten bleibt. Um die Distanz zwischen den Unterbäumen  $T_{v_i}$  und  $T_{v_k}$  zu berechnen, werden die Richtungskonturen  $C_{v_i, -\vec{D}_{i,k}}$  und  $C_{v_k, \vec{D}_{i,k}}$  verwendet. Ist die Distanz zwischen diesen ist  $\minDist = 10$ . Weil  $|\vec{D}_{i,k}| = 2$ , ergibt sich bei Betrachtung dieser beiden Unteräume als maximal möglicher Verkürzungswert  $\left\lfloor \frac{(\minDist-1)}{|\vec{D}_{i,k}|} \right\rfloor = \left\lfloor \frac{(10-1)}{2} \right\rfloor = 4$ .

Um Überlagerungen zwischen Unterbäumen der Kinder von  $v$  mit den eingehenden Kanten der Geschwisterknoten zu vermeiden, müssen diese Distanzen ebenso beachtet werden. In diesem Fall wird  $\minDist$  zwischen allen Paaren von inzidenten Kanten  $(v, v_i)$  von  $v$  und den Richtungskonturen  $C_{v_j, \vec{D}_j}$  mit  $\vec{D}_j = \frac{\vec{v}_j}{|\vec{v}_j|}$  bestimmt. Die Kante  $(v, v_j)$  kann dann um den Wert  $\minDist - 1$  verkürzt werden. In Abbildung 4.13(a) ist der Abstand zwischen  $C_{v_i, \vec{D}_i}$  zur Kante  $(v, v_h)$  mit sechs eingezeichnet. Somit kann  $(v, v_h)$  höchstens um fünf verkürzt werden, ohne die Planarität zu zerstören.

**Lemma 4.2.** *Seien  $v_1, \dots, v_d$  Kinder von  $v$ . Dann werden in der Distanzberechnung der Kontur  $C_{v_i}$  nur Richtungskonturen  $C_{v_i, \vec{D}}$  benötigt mit  $\vec{D} \in \{(v_i, v_j) \mid v_j \text{ ist Geschwisterknoten von } v_i\}$ .*

Dazu reicht es Abbildungen 4.3, 4.4 und 4.5 zu betrachten, in der die Menge der Bewegungsrichtungen auf dem  $k$ -Gitter für  $k \in \{4, 6, 8\}$  eingezeichnet sind.

**Lemma 4.3.** *Sei  $v$  ein Knoten mit den im Rekursionsschritt zurückgegebenen Konturen  $C_{v_1}, \dots, C_{v_d}$  der Kinder  $v_1, \dots, v_d$ . Die in der Distanzberechnung eines Knotens  $v_i$  mit  $i \in \{1, \dots, d\}$  benötigten Richtungskonturen  $C_{v_i, \vec{D}}$  mit  $\vec{D} \in \{(v_i, v_j) \mid v_j \text{ ist Geschwisterknoten von } v_i\}$  verdecken die Kante  $(v, v_i)$  nicht vollständig.*

*Beweis.* Betrachtet man die relativen Bewegungsrichtungen der Kindknoten von  $u$  paarweise, so ergeben sich die Richtungsvektoren, wie in Abbildungen 4.3, 4.4 und 4.5. Die Menge  $R$  dieser Richtungskonturen der Richtungsvektoren werden für die Berechnung der Distanz zwischen gegenüberliegenden Konturen benötigt. Initial werden die Kantenlängen der ausgehenden Kanten des Knotens  $u$  durch die Formel  $3^{h(T)-t(u)-1}$  im Baum  $T$  gesetzt.

Da die Summe der Kantenlängen jedes Pfades  $(v, \dots, b)$  eines Kindes  $v$  von  $u$  zu jedem Blatt  $b$  kleiner ist als die Länge  $3^{h(T)-t(u)-1}$  der ausgehenden Kante  $(u, v)$ , können deren Richtungskonturen die ausgehenden Kanten von  $u$  nicht verdecken. Somit ist in der initialen Zeichnung die eingehende Kante  $(u, v)$  von  $v$  nicht durch die entsprechenden Richtungskonturen der Menge  $R$  verdeckt, siehe Abbildung 3.27.  $\square$

Schließlich ist der endgültige Verkürzungswert das Minimum aller berechneten Verkürzungswerte und die ausgehenden Kanten von  $v$  können um diesen verkürzt werden. In Abbildung 4.13(a) können die ausgehenden Kanten maximal um vier verkürzt werden. Das Resultat der Verkürzung ist in Abbildung 4.13(b) dargestellt. Als nächstes werden die gegebenen verschobenen Richtungskonturen zu einer neuen Gesamtkontur  $C_v$  für den Knoten  $v$  vereinigt.

#### 4.2.2.2 Vereinigen der Konturen

Nach dem Verkürzen der ausgehenden Kanten  $(v_1, \dots, v_i)$  von  $v$  in der Zeichnung  $\Gamma_k(T_v)$  wird die Kontur  $C_v$  des Unterbaumes  $T_v$  von  $v$  und damit die entsprechenden Richtungskonturen  $C_{v, \vec{D}}$  für die Richtungen  $\vec{D} \in \mathcal{D}_k$  berechnet.

Zuerst wird die triviale Richtungskontur  $C_{v, \vec{D}}$ , die in Zeile 1 in Algorithmus 4.3 initialisiert wurde, mit den Konturen der ausgehenden Kanten  $(v, v_1), \dots, (v, v_i)$  zusammengeführt. Daraus ergibt sich die Richtungskontur der ausgehenden Kanten  $(v, v_1), \dots, (v, v_i)$ , siehe dazu Abbildung 4.13(b). Im nächsten Schritt wird eine Kontur  $C_{v_i, \vec{D}} \in \mathcal{C}$  ( $i \in \{1, \dots, d\}$ ) mit  $C_{v, \vec{D}}$  vereinigt und aus  $\mathcal{C}$  entfernt. Dies wird solange durchgeführt bis  $\mathcal{C}$  leer ist. Die Kontur  $C_{v, \vec{D}}$  ist immer zusammenhängend, da zuerst die Richtungskontur der ausgehenden Kanten erzeugt wird und die ausgehenden Kanten alle inzident zu  $v$  sind. Eine Richtungskontur  $C_{v_i}$  mit  $i \in \{1, \dots, d\} \in \mathcal{C}$  ist immer eine Richtungskontur des Unterbaumes  $T_{v_i}$ , dessen Wurzel  $v_i$  bereits von der Kontur  $C_{v, \vec{D}}$  verdeckt wird. Und da der Unterbaum zusammenhängend ist, ist die aktualisierte Kontur wieder zusammenhängend. Die neue Kontur  $C_{v, \vec{D}}$  beinhaltet alle sichtbaren Segmente von  $C_{v_1, \vec{D}}, \dots, C_{v_i, \vec{D}}$  und sichtbaren Segmente der ausgehenden Kanten von  $v$ , siehe Abbildung 4.13(b). Richtungskonturen die nicht mehr Teil von  $C_{v, \vec{D}}$  sind, wie z. B.  $C_{v_j, \vec{D}}$  in Abbildung 4.13(b), werden in den weiteren Rekursionsschritten nicht mehr aus der Richtung  $\vec{D}$  sichtbar sein. Somit werden diese Teile sicher nicht mehr zu einer Distanzberechnung benötigt und daher nicht mehr berücksichtigt. Eine kompaktere Zeichnung des Baumes aus Abbildung 4.9 ist in Abbildung 4.14 dargestellt, welche mit Algorithmus 4.1 unter Verwendung von Algorithmus 4.3 gezeichnet wurde.

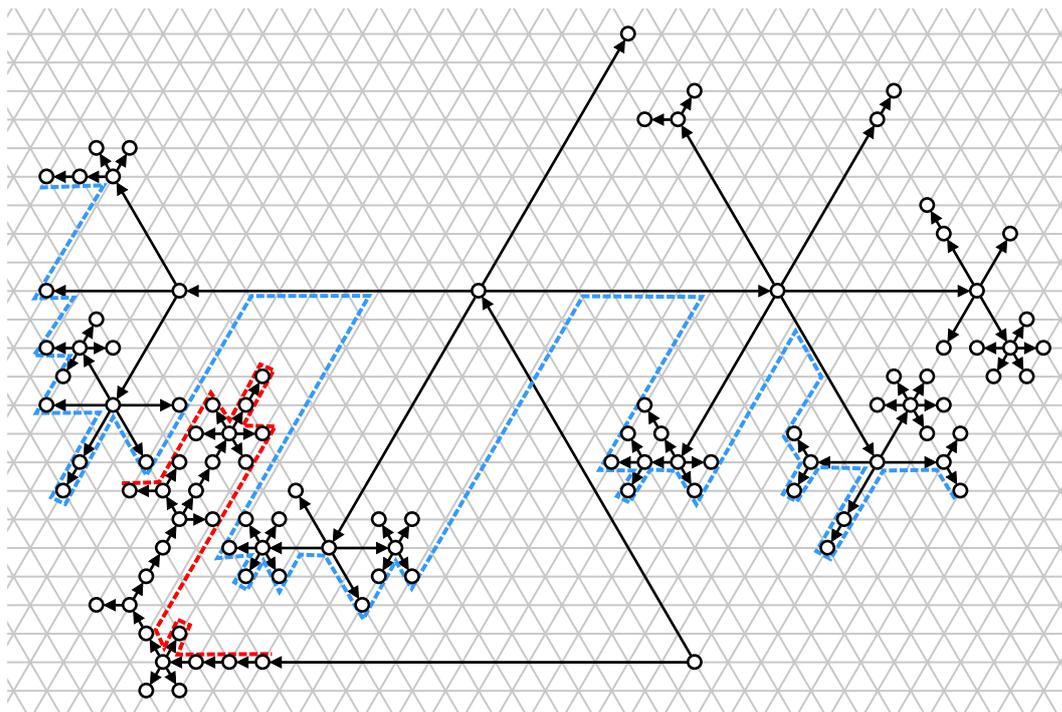


Abbildung 4.14. Negativbeispiel Algorithmus 4.2

### 4.2.3 Zeitkomplexität

Die beschriebene Berechnung von ordnungserhaltenden lokal uniformen Patternzeichnungen mit Algorithmus 4.3 hat eine Zeitkomplexität in  $\mathcal{O}(n^2)$ . In jedem Rekursionsschritt ist die Laufzeit abhängig von den Scanline-Verfahren des Verkürzungsschrittes (Abschnitt 4.2.2.1) und des Vereinigungsschrittes (Abschnitt 4.2.2.2). Aufgrund der Linearzeit, die für die Scanline-Verfahren benötigt wird, ergibt sich für jeden rekursiven Aufruf eine Laufzeit in  $\mathcal{O}(n)$ . Im Vereinigungsschritt werden paarweise die Richtungskonturen zusammengeführt. Dieses Zusammenführen ist für jedes Paar von Richtungskonturen mit einem Scanline-Verfahren auch in  $\mathcal{O}(n)$  durchführbar. Die maximale Anzahl an Richtungskonturen pro Richtung  $\vec{D} \in \mathcal{D}_k$  ist die Anzahl der Kinder von  $v$ , also für  $k \in \{4, 6, 8\}$  in  $\mathcal{O}(k)$ . Folglich benötigt das Vereinigen der Konturen lineare Zeit. Der rekursive Aufruf wird höchstens einmal für jeden Knoten ausgeführt, weshalb sich eine Gesamtlaufzeit in  $\mathcal{O}(n^2)$  ergibt.

**Satz 4.3.** *Der Algorithmus 4.1 mit Verwendung von Algorithmus 4.3 erzeugt kompaktierte  $OLP_k$ -Zeichnungen mit den Ästhetikkriterien Planarität, Ganzzahligkeit und identische Zeichnungen für isomorphe Unterbäume bis auf Rotation und Translation in  $\mathcal{O}(n^2)$ -Zeit.*

Somit ist es möglich Zeichnungen in quadratischer Zeit mit einem relativ einfachen Verfahren zu erstellen. Es ist jedoch auch interessant, ob ein Verfahren existiert,

das dieselben Zeichnungen in Linearzeit erzeugt. Ein solches Verfahren wird im folgenden Abschnitt präsentiert.

### 4.3 Zeichnungen von $d$ -nären Bäumen ohne Unterbaum Separation in Linearzeit

Mit dem Ablaufen der Konturen mit den beschriebenen Scanline-Verfahren im Verkürzungsschritt und Vereinigungsschritt ist die Laufzeit des vorgestellten Verfahrens Algorithmus 4.3 in  $\mathcal{O}(n^2)$ . Wenn, wie im Algorithmus von Reingold und Tilford, die Konturen nicht beliebig abgelaufen werden, kann Linearzeit erreicht werden [Wal90, RT81, B JL02]. Die Problemstellen stellen die Distanzberechnung zwischen Konturen und der Vereinigungsschritt im Rekursionsschritt dar. Eine Linearzeit-Implementierung ist im Graph Editor Toolkit *Gravisto* [BBF<sup>+</sup>04] realisiert.

Im Rekursionsschritt dürfen nur Segmente von Konturen bzw. Richtungskonturen abgelaufen werden, die für die restlichen Rekursionsschritte nicht mehr benötigt werden. Diese Forderung ist notwendig um die Linearzeit des Gesamtalgorithmus zu gewährleisten. Dazu werden im folgenden Abschnitt Definitionen und Vorarbeiten beschrieben, um danach das Verfahren und die Anpassung dieser Problemstellen zu erläutern.

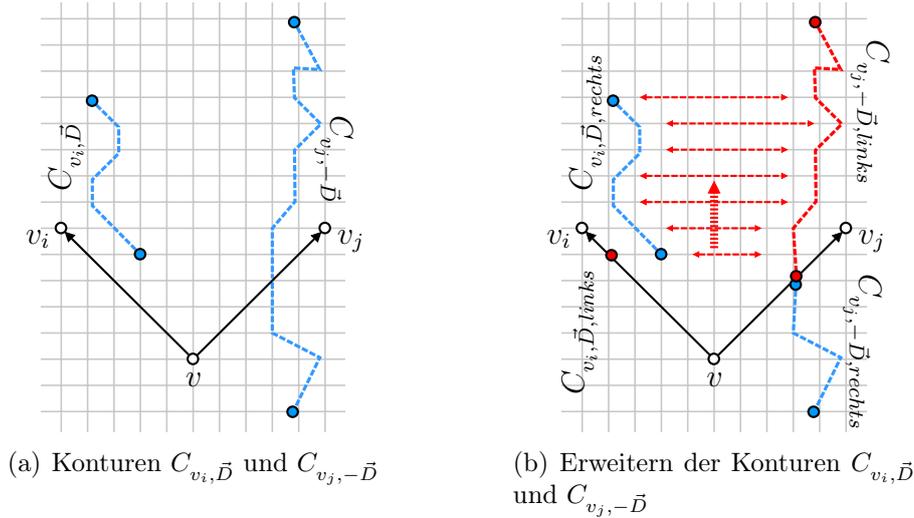
#### 4.3.1 Definitionen

Um in der Distanzberechnung und im Vereinigungsschritt nicht von den Endpunkten einer Kontur starten zu müssen, werden die Konturen in zwei Teile zerlegt. Diese Aufteilung stellt sicher, dass immer ein Endpunkt dieser Teile verdeckt ist und im Forlauf der Rekursion nicht mehr benötigt wird.

Für jedes Kind  $v_i$  von  $v$  gibt der rekursive Aufruf in Algorithmus 4.3 die Kontur  $C_{v_i}$ , also die  $|\mathcal{D}_k|$  Richtungskonturen  $C_{v_i, \vec{D}}$  mit  $\vec{D} \in \mathcal{D}_k$ , zurück. Dabei wird jeder Richtungskontur  $C_{v_i, \vec{D}}$  ein *Typ* 1, 2a oder 2b zugeordnet. Sei  $C_{v_i, \vec{D}}$  eine Richtungskontur des Knotens  $v_i$  (Wurzel des Unterbaumes  $T_{v_i}$ ) mit der Richtung  $\vec{D} \in \mathcal{D}_k$ .

1. Wenn  $C_{v_i, \vec{D}}$  den Vaterknoten  $v$  von  $v_i$  in Richtung  $\vec{D}$  vdeckt, ist die Richtungskontur vom *Typ* 1.
2. Andernfalls ist sie vom *Typ* 2.
  - (a) Wenn  $C_{v_i, \vec{D}}$  die Kante  $(v, v_i)$  schneidet, ist sie vom *Typ* 2a.
  - (b) Andernfalls, wenn  $C_{v_i, \vec{D}}$  die Kante  $(v, v_i)$  nicht schneidet, so ist sie vom *Typ* 2b.

Eine Kontur  $C_{v_i, \vec{D}}$  vom *Typ* 2a oder 2b wird in zwei Teile zerlegt, siehe Abbildung 4.15. Sei  $L_i$  ( $R_i$ ) der linke (rechte) Endpunkt von  $C_{v_i, \vec{D}}$ .



**Abbildung 4.15.** Erweitern der Konturen in der Distanzberechnung

Eine Typ 2a Kontur ist aufgeteilt in eine *linke Hälfte*  $C_{v_i, \vec{D}, links}$  von  $L_i$  bis zum Schnittpunkt  $x$  der Kontur  $C_{v_i, \vec{D}}$  mit der Kante  $(v, v_i)$ . Die *rechte Hälfte*  $C_{v_i, \vec{D}, rechts}$  ist der entsprechende Teil von  $x$  bis zu  $R_i$ . Der Schnittpunkt der Kontur mit der eingehenden Kante von  $v$  wird aus dem vorigen Rekursionsschritt aus dem Vereinigungsschritt mitgegeben, um den Schnittpunkt nicht mit einem Durchlauf der kompletten Liste suchen zu müssen. Das effiziente Bestimmen des Schnittpunktes wird in Abschnitt 4.3.2.3 beschrieben.

Für Typ 2b Konturen schneidet entweder die Projektion des linken Endpunkt  $L_i$  von  $C_{v_i, \vec{D}}$  in Richtung  $\vec{D}$  den Punkt  $P_x$  auf der Kante  $(v, v_i)$  oder die Projektion des rechten Endpunktes  $R_i$ . O. B. d. A. schneidet die Projektion von  $L_i$  die Kante  $(v, v_i)$ . Der Schnittpunkt  $x$  wird als linke Hälfte  $C_{v_i, \vec{D}, links}$  bezeichnet und die ursprüngliche Kontur  $C_{v_i, \vec{D}}$  als  $C_{v_i, \vec{D}, rechts}$ . Schneidet die Projektion von  $R_i$  die Kante, dann wird umgekehrt verfahren, siehe Abbildung 4.15(b).

### 4.3.2 Beschreibung des Verfahrens

Der Algorithmus ist im Grundprinzip genau so aufgebaut wie Algorithmus 4.3 mit Unterschieden in der Distanzberechnung und im Vereinigungsschritt. In beiden Teilen dürfen Konturen nicht mehr beliebig abgelaufen werden, sondern nur noch die Teile von Konturen, die in zukünftigen Rekursionsschritten nicht mehr benötigt werden, die also wegfallen.

In der Distanzberechnung mit den Konturen  $C_{v_i, \vec{D}_{i,j}}$  und  $C_{v_j, -\vec{D}_{i,j}}$  fallen die Teile der Konturen sicher weg, die sich entlang  $\vec{D}_{i,j}$  bzw.  $-\vec{D}_{i,j}$  gegenüberliegen.

Im Vereinigungsschritt werden die Teile der Konturen nicht mehr benötigt, durch die andere Teile der Konturen in der Richtung  $\vec{D} \in \mathcal{D}_k$  verdeckt werden. Im Folgen-

den werden die Details im Verkürzungs- und Vereinigungsschritt beschrieben.

### 4.3.2.1 Verkürzen der Kanten

Das Verkürzen der ausgehenden Kanten im Rekursionsschritt verläuft analog zum Verkürzungsschritt in Algorithmus 4.3 mit dem Unterschied, dass nur noch Teile im Scanline-Verfahren abgelaufen werden, die nach dem aktuellen Rekursionsschritt wegfallen.

Betrachte dazu die Konturen von  $v_i$  und  $v_j$  mit dem Vater  $v$  in Richtung  $\vec{D}_{i,j}$  bzw.  $-\vec{D}_{i,j}$ . Nach Lemma 4.3 sind die Hälften  $C_{v_i, \vec{D}_{i,j}, links}$  und  $C_{v_j, -\vec{D}_{i,j}, rechts}$  am Projektionspunkt  $P_x$  bzw. am Schnittpunkt  $x$  getrennt worden. Dann werden diese dort beginnend simultan abgelaufen und der minimale Abstand entlang  $\vec{D}$  bestimmt, siehe Abbildung 4.15(b). Der simultane Ablauf stoppt mit dem Ende der ersten Kontur, vgl. Algorithmus von Reingold und Tilford [RT81]. Die anderen beiden Hälften  $C_{v_i, \vec{D}, rechts}$  und  $C_{v_j, -\vec{D}, links}$  werden ebenfalls startend am Teilpunkt abgelaufen und die minimale Distanz bestimmt, siehe Abbildung 4.15(b).

Wie auch in Algorithmus 4.3 müssen danach die Abstände zu den ausgehenden Kanten von  $v$  berücksichtigt werden, damit die Unterbäume diese nicht überlagern. Die übrige Distanzberechnung erfolgt analog zum Verfahren in Algorithmus 4.3, so dass die Kanten um den berechneten Minimalwert verkürzt werden können. Nach dieser Verkürzung werden die verschobenen Konturen vereinigt, was im nachfolgenden Abschnitt beschrieben ist. Wichtig ist, dass das Scanline-Verfahren in der Distanzberechnung am Schnittpunkt  $x$  oder Projektionspunkt  $P_x$  startet, da diese beiden Anfangsstücke am Ende des Rekursionsschrittes nach dem Durchlauf wegfallen.

### 4.3.2.2 Vereinigen von Konturen

Der letzte Schritt im Verfahren ist die Vereinigung der Konturen. Die Ausführung ist wieder ähnlich zu Algorithmus 4.3. Es müssen die Richtungskonturen  $C_{v_i, \vec{D}}$  der  $d$  Kinder  $v_i$  mit  $i \in \{1, \dots, d\}$  von  $v$  zu einer Kontur  $C_{v, \vec{D}}$  vereinigt werden. Die Richtung ist dabei durch  $\vec{D} \in \mathcal{D}_k$  vorgegeben.

Jede Richtungskontur  $C_{v_i, \vec{D}}$  eines Kindknotens  $v_i$  von  $v$  für die Richtung  $\vec{D} \in \mathcal{D}_k$  wird zuerst mit der eingehenden Kante  $(v, v_i)$  vereinigt. Um hier nur die Teile abzulaufen, die verdeckt sind, werden die linke Hälfte  $C_{v_i, \vec{D}, links}$  und die rechte Hälfte  $C_{v_i, \vec{D}, rechts}$  verwendet. Der Endpunkt einer der beiden Hälften fällt mit der Kante  $(v, v_i)$  zusammen, siehe dazu die Definition von linker und rechter Hälfte in Abschnitt 4.3.1. Damit kann in konstanter Zeit überprüft werden, welcher Teil durch  $(v, v_i)$  verdeckt wird, um diesen dann durch die Kante zu ersetzen. Sind die Hälften der Richtungskonturen  $C_{v_i, \vec{D}, links}$  und  $C_{v_i, \vec{D}, rechts}$  mit der Kante  $(v, v_i)$  vereinigt, dann können die Richtungskonturen paarweise vereinigt werden.

Projiziert man die Endpunkte  $L_i, R_i, L_j, R_j$  zweier Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  entlang  $\vec{D}$  auf die Gerade  $\vec{D}^o$ , dann ergeben sich die Reihenfolgen der Projektionspunkte  $(L_i, L_j, R_i, R_j)$  und  $(L_i, L_j, R_j, R_i)$ , siehe Abbildung 4.16 und 4.17.

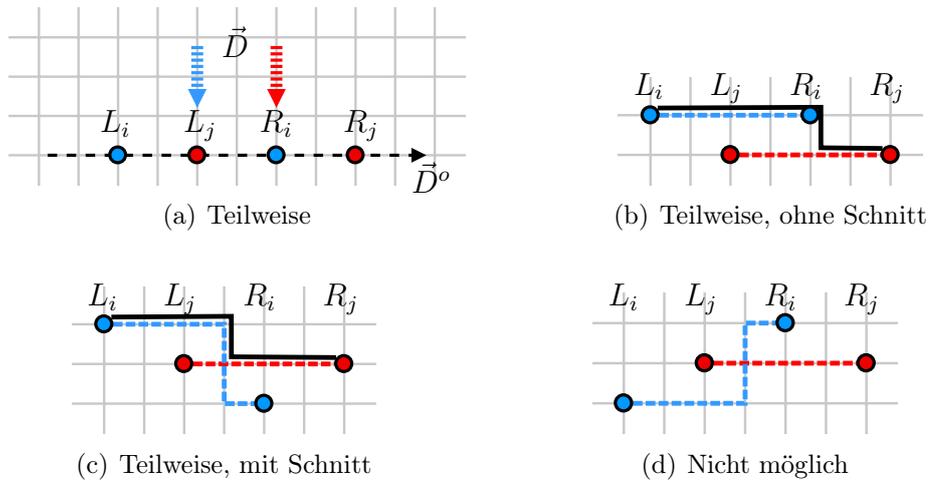


Abbildung 4.16. Fälle im Vereinigungsschritt für die Reihenfolge  $(L_i, L_j, R_i, R_j)$  der Projektion in Richtung  $\vec{D} = (0, 1)$

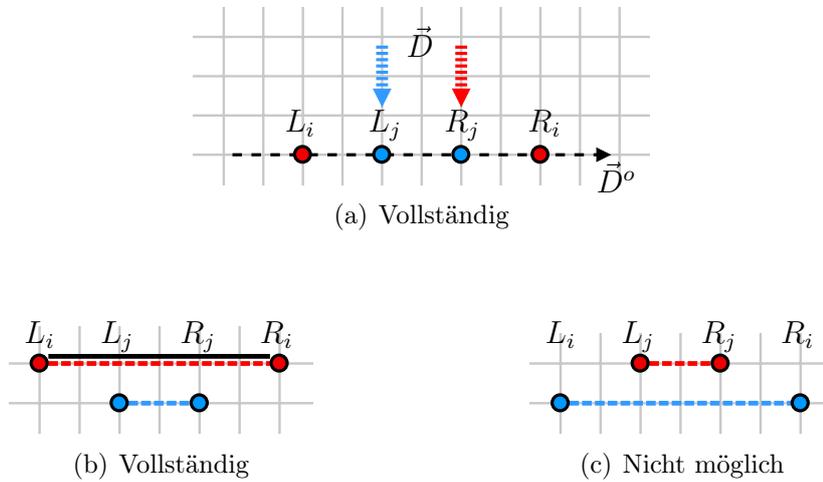


Abbildung 4.17. Fälle im Vereinigungsschritt für die Reihenfolge  $(L_i, L_j, R_j, R_i)$  der Projektion in Richtung  $\vec{D} = (0, 1)$

Zwei Richtungskonturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  verdecken sich entweder *teilweise* (Abbildung 4.16) oder *vollständig* (Abbildung 4.17) aus der Richtung  $\vec{D}$ .

Eine teilweise Überlagerung, wie in Abbildung 4.17(c) gezeigt, ist nicht möglich, da Pfade von  $v_i$  zu einem Endpunkte  $L_i$  oder  $R_i$  der Richtungskontur  $C_{v_i, \vec{D}}$  existieren müssten, die den Unterbaum  $T_{v_j}$  nicht schneiden. Folglich müsste der Pfad um den Unterbaum  $T_{v_j}$  herum verlaufen, was aber dann zur Folge hätte, dass die Kontur  $C_{v_i, \vec{D}}$  nicht die Endpunkte  $L_i$  bzw.  $R_i$  hätte.

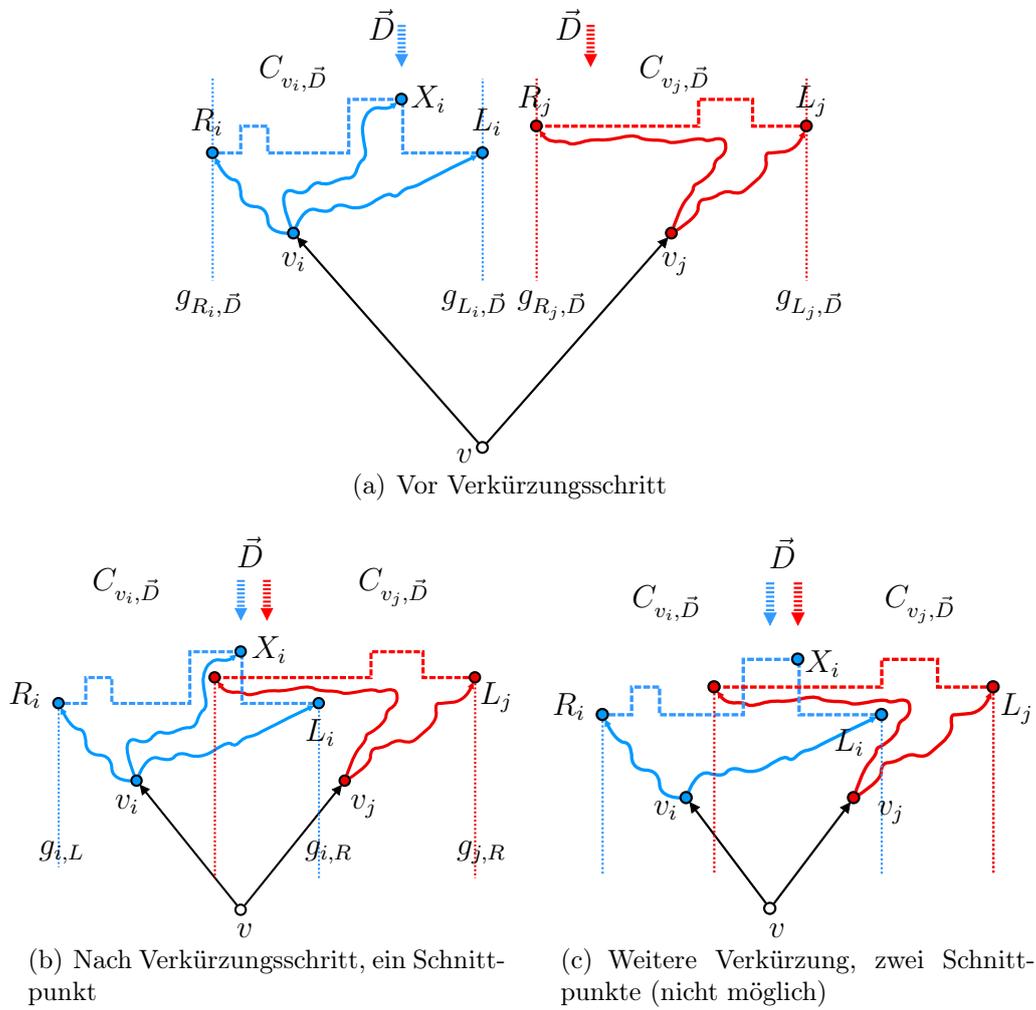
Nach dem Verkürzen der ausgehenden Kanten  $(v, v_i)$  und  $(v, v_j)$  von  $v$  können sich die Richtungskonturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  schneiden, siehe 4.16(c). Das Vereinigen zweier Richtungskonturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$ , die sich teilweise überlagern, vollständig überlagern oder schneiden, kann mit dem Scanline-Verfahren durchgeführt werden (siehe Abschnitt 4.3.2.2). Hierbei werden die beiden Richtungskonturen entlang  $\vec{D}^\circ$  abgelaufen. Dazu muss der Endpunkt bestimmt werden, der sicher durch die andere Richtungskontur verdeckt ist, um nur verdeckte Teile abzulaufen.

Im Verfahren in Abschnitt 4.2.2.2 war es nicht notwendig diese Fälle zu unterscheiden, da das Scanline-Verfahren im Vereinigungsschritt alle Segmente der beteiligten Richtungskonturen abgelaufen ist. Dies hat zur Folge, dass jeder Rekursionsschritt eine Zeitkomplexität in  $\mathcal{O}(n)$  und somit der gesamte Algorithmus eine Zeitkomplexität in  $\mathcal{O}(n^2)$  besitzt. In diesem Abschnitt soll die Laufzeit des Gesamtalgorithmus auf  $\mathcal{O}(n)$  verbessert werden. Wie auch im Verkürzungsschritt dürfen nur Segmente abgelaufen werden, die im nächsten Rekursionsschritt nicht mehr Teil einer Kontur sind.

An diesem Punkt stellt sich die Frage, ob überhaupt Endpunkte existieren, die wegfallen. Betrachte dazu die folgende Fallunterscheidung von Konturpaaren  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  bezüglich der Richtung  $\vec{D} \in \mathcal{D}_k$  in Abbildungen 4.16 und 4.17. Die Endpunkte werden auf eine Gerade parallel zu  $\vec{D}^\circ$  in Richtung  $\vec{D}$  projiziert, so dass sich Reihenfolgen für die Endpunkte ergeben. Die Projektion des linken Endpunktes von  $C_{v_i, \vec{D}}$  ( $C_{v_j, \vec{D}}$ ) wird mit  $L_i$  ( $L_j$ ) und die Rechte mit  $R_i$  ( $R_j$ ) bezeichnet. Diese Projektionspunkte können verschiedene Reihenfolgen haben, wobei der linke Projektionspunkt nach Definition links vom Rechten ist. Somit ergeben sich o. B. d. A. die Reihenfolgen  $(L_i, L_j, R_i, R_j)$  in Abbildung 4.16(a) und  $(L_i, L_j, R_j, R_i)$  in Abbildung 4.17(a) oder umgekehrt. Da sich die Konturen mindestens an einem Punkt hinsichtlich des Richtungsvektors  $\vec{D}$  überlappen, ist Reihenfolge  $(L_i, R_i, L_j, R_j)$  nur mit  $R_i = L_j$  möglich und somit durch  $(L_i, L_j, R_i, R_j)$  abgedeckt.

Betrachte zuerst den Fall  $(L_i, L_j, R_i, R_j)$  mit  $L_i < L_j < R_i < R_j$  entlang der Richtung  $\vec{D}^\circ$ , siehe Abbildung 4.16(a). Gesucht ist ein Endpunkt der Konturen, der von der anderen Kontur verdeckt wird. Die beiden äußeren Endpunkte  $L_i$  und  $R_j$  sind nach dem Vereinigungsschritt sicher Teil der neuen Kontur, da sie außen liegen. Das Konturpaar kann in diesem Fall verschiedene Eigenschaften annehmen.

**Lemma 4.4.** *Nach der Verkürzung der ausgehenden Kanten  $(v, v_i)$  und  $(v, v_j)$  können sich zwei Richtungskonturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  von Geschwisterknoten  $v_i$  und  $v_j$  mit  $\vec{D} \in \mathcal{D}_k$  schneiden.*



**Abbildung 4.18.** Verkürzen der ausgehenden Kanten und resultierende Konturen  $\vec{D} = (0, 1)$  und  $\vec{D}^o = (1, 0)$

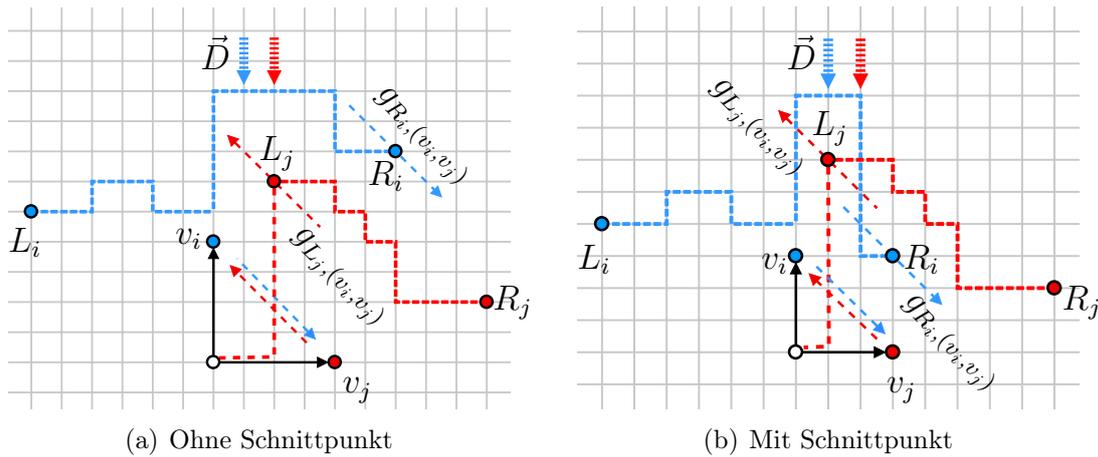
*Beweis.* Seien  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  zwei Richtungskonturen der Kinder  $v_i$  und  $v_j$  von  $v$  in Richtung  $\vec{D}$ . O. B. d. A. ist  $\vec{D} = (0, 1)$ , wie in Abbildung 4.19. Dann kann oberhalb der Kontur  $C_{v_i, \vec{D}}$  kein Knoten oder keine Kante des Unterbaumes  $T_{v_i}$  gezeichnet sein (analog für  $C_{v_j, \vec{D}}$ ). Außerdem kann es nicht sein, dass ein Knoten oder eine Kante von  $T_{v_i}$  links (rechts) von der zu  $\vec{D}$  parallelen Gerade  $g_{L_i, \vec{D}}$  ( $g_{R_i, \vec{D}}$ ) platziert ist. Dasselbe gilt für den Unterbaum  $T_{v_j}$ . Angenommen,  $X_i$  ist der Knoten, der den oberen Endpunkt des zu  $\vec{D}$  parallelen Segments darstellt, das den Schnitt erzeugt. Dabei ist zu beachten, dass dieses Segment nicht auf einer Kante liegen kann, da sonst die Planarität verletzt wäre. Da der Baum zusammenhängend ist, muss ein Pfad zum Knoten  $X_i$  existieren, der die Geraden  $g_{L_i, \vec{D}}$  und  $g_{R_i, \vec{D}}$  nicht schneidet. Somit ist es möglich, dass ein Schnitt zwischen Konturen im Verkürzungsschritt erzeugt wurde.  $\square$

Als Beispielskizze ist in Abbildung 4.18(b) gezeigt, dass Konturen sich schneiden können. Die ursprüngliche Zeichnung vor dem Verkürzungsschritt ist in Abbildung 4.18(a) dargestellt. Werden die ausgehenden Kanten von  $v$  in Abbildung 4.18(b) noch weiter verkürzt, so ergibt sich die Zeichnung Abbildung 4.18(c). Diese Zeichnung kann jedoch nicht zustande kommen, da sie zwei Schnittpunkte zwischen den Konturen besitzt und somit nicht mehr planar und zusammenhängend sein kann.

**Lemma 4.5.** *Nach der Verkürzung der ausgehenden Kanten  $(v, v_i)$  und  $(v, v_j)$  haben zwei Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  mit  $\vec{D} \in \mathcal{D}_k$  höchstens einen Schnittpunkt.*

*Beweis.* Gegeben sei  $\Gamma(T_v)$  eine planare Zeichnung nach dem Verkürzen der ausgehenden Kanten  $(v, v_i)$  und  $(v, v_j)$  von  $v$ . Angenommen,  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  besitzen zwei Schnittpunkte, wie beispielsweise in Abbildung 4.18(c). Dann gilt, dass überhalb einer Kontur in Richtung  $\vec{D}$  links von  $L_i$  und rechts von  $R_i$  keine Knoten und Kanten platziert sein können. Betrachte die Pfade zu den Endpunkten  $L_i, R_i, L_j$  und  $R_j$  der beiden Konturen. Da auf den Koordinaten der Endpunkte einer Kontur  $C_{v_i, \vec{D}}$  Knoten platziert sein müssen, existiert wegen des Zusammenhangs ein Pfad  $(v_i, \dots, L_i)$ ,  $(v_i, \dots, R_i)$ ,  $(v_j, \dots, L_j)$  und  $(v_j, \dots, R_j)$ . Auf den Koordinaten von  $X_i$  (vgl. Beweis von Lemma 4.4) muss ebenfalls ein Knoten platziert sein, da ein Knick nur durch einen Knoten hervorgerufen werden kann. Somit muss ebenfalls ein Pfad  $(v_i, \dots, X_i)$  existieren. Dieser Pfad kann aber in einer planaren Zeichnung nicht existieren, da er den Pfad  $(v_j, \dots, R_j)$  schneiden müsste, so dass dies zu einem Widerspruch führt.  $\square$

Somit ergeben sich zwei zu unterscheidende Fälle, nämlich entweder schneiden zwei Konturen sich nicht oder sie schneiden sich genau einmal. Wenn sich zwei Konturen einmal schneiden, dann kann nur der Fall in Abbildung 4.16(c) auftreten. Der gezeigte Fall in Abbildung 4.16(d) ist nicht möglich. Dazu können dieselben Argumente verwendet werden, wie im Beweis von Lemma 4.5. Es kann aufgrund des Zusammenhangs kein Pfad zum Knoten  $R_i$  existieren ohne die Planarität zu verletzen. Also gibt es bis auf Spiegelung der Reihenfolge nur zwei Möglichkeiten wie zwei Konturen zueinander platziert sein können, wenn die Reihenfolge  $(L_i, L_j, R_i, R_j)$



(a) Ohne Schnittpunkt

(b) Mit Schnittpunkt

**Abbildung 4.19.** Vereinigung zweier Konturen mit teilweiser Verdeckung  $(L_i, L_j, R_i, R_j)$  nach dem Verkürzungsschritt

ist. Siehe dazu die Abbildungen 4.16(b) und 4.16(c) in denen die Vereinigung der beiden Konturen ebenfalls eingezeichnet ist.

Für die Reihenfolge  $(L_i, L_j, R_j, R_i)$  mit  $L_i < L_j < R_j < R_i$  sind die äußeren Endpunkte  $L_i$  und  $R_i$  ebenfalls Teil der Vereinigung der Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_i, \vec{D}}$ . Somit stellt sich nur die Frage, ob die mittlere Kontur  $C_{v_i, \vec{D}}$  komplett verdeckt ist, ob sie die Kontur  $C_{v_i, \vec{D}}$  teilweise verdeckt, oder ob es einen Schnitt der beiden Konturen geben kann.

Angenommen,  $C_{v_j, \vec{D}}$  liegt überhalb von  $C_{v_i, \vec{D}}$ , siehe Abbildung 4.17(c). Dann müsste ein Pfad zu einem der beiden Eckpunkte  $L_j$  oder  $R_j$  existieren. Dies ist aufgrund derselben Argumentation mit Planarität und Zusammenhang nicht möglich. Damit kann dieser Fall ausgeschlossen werden. Für den Schnitt der beiden Konturen führt dieselbe Argumentation zum Widerspruch.

Somit bleibt nur noch der Fall, dass die Kontur  $C_{v_i, \vec{D}}$  die Kontur  $C_{v_j, \vec{D}}$  komplett verdeckt, siehe Abbildung 4.17(b). Sind die projizierten Koordinaten zweier Endpunkte  $R_i$  und  $R_j$  ( $L_i$  und  $L_j$ ) gleich auf, so kann analog verfahren werden.

Jetzt stellt sich die Frage, wie die Vereinigung von Konturen dieser möglichen Fälle funktioniert, ohne dabei nicht wegfallende Kontursegmente abzulaufen. Zuerst wird der Fall  $(L_i, L_j, R_i, R_j)$  betrachtet, wo sich zwei Konturen entweder schneiden oder nicht schneiden, siehe Abbildungen 4.16(b) und 4.16(c).

Betrachte zwei Richtungskonturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  mit den Unterbäumen der Geschwisterknoten  $v_i$  und  $v_j$  und den entsprechenden Endpunkten  $L_i, R_i, L_j, R_j$  für die Richtung  $\vec{D}$ , die sich teilweise überlagern. Somit ist mindestens ein Endpunkt von  $R_i$  und  $L_j$  durch die andere Kontur verdeckt. Sei die Gerade  $g_{P, (v_i, v_j)}$  die Gerade die parallel zur Richtung  $(v_i, v_j)$  verläuft und den Punkt  $P$  schneidet. Entlang dieser Richtung bewegen sich die Unterbäume relativ zueinander und somit die Konturen, wenn die ausgehenden Kanten  $(v, v_i)$  und  $(v, v_j)$  verkürzt werden. Wenn die Rich-

tung  $(v_i, v_j)$  ersichtlich ist, wird die Gerade einfach  $g_P$  abgekürzt. Somit sind die zwei Geraden  $g_{L_i, (v_i, v_j)}$  und  $g_{R_j, (v_i, v_j)}$  parallel zur Bewegungsrichtung  $(v_i, v_j)$ . Beim Verkürzen der ausgehenden Kanten von  $v$  verschiebt sich der Endpunkt  $L_i$  ( $R_j$ ) exakt auf der Geraden  $g_{L_i, (v_i, v_j)}$  ( $g_{R_j, (v_i, v_j)}$ ). Der *tieferer Endpunkt bezüglich  $v$*  ist der Endpunkt der Konturen, dessen Gerade den kleineren Abstand zur Koordinate des Knotens  $v$  besitzt, siehe Abbildung 4.19. In Abbildung 4.19(a) ist  $L_j$  der tiefere Endpunkt, da die Gerade  $g_{L_i, (v_i, v_j)}$  einen kleineren Abstand zu  $v$  besitzt als die Gerade  $g_{R_j, (v_i, v_j)}$ . In Abbildung 4.19(b) ist es andersherum.

**Lemma 4.6.** *Verdecken sich zwei Konturen nach der Verkürzung der Kanten teilweise (mit oder ohne Schnittpunkt) in der Form  $(L_i, L_j, R_i, R_j)$ , dann ist der tiefere Endpunkt von  $L_j$  ( $R_i$ ) durch die Kontur  $C_{v_i, \vec{D}}$  ( $C_{v_j, \vec{D}}$ ) in Richtung  $\vec{D}$  verdeckt.*

*Beweis.* Betrachte zwei Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  mit den entsprechenden Endpunkten  $L_i, R_i, L_j, R_j$ . Dann ist der Endpunkt verdeckt, dessen Gerade den kleineren Abstand zu  $v$  besitzt. O. B. d. A. hat  $g_{L_i}$  den geringeren Abstand zu  $v$ . Dann muss der Endpunkt  $L_i$  beim Verkürzen unter die andere Kontur geschoben worden sein und ist somit entlang der Konturrichtung  $\vec{D}$  verdeckt.  $\square$

Da mit Lemma 4.6 klar ist, welcher der beiden Endpunkte  $L_j$  und  $R_i$  verdeckt ist, kann das Scanline-Verfahren an diesem Punkt beginnen. Da sich die beiden zu vereinigenden Konturen entweder schneiden oder nicht, wird ein Verfahren präsentiert, das für beide Möglichkeiten eine Vereinigung der Konturen mit der Bedingung erstellt, nur die verdeckten Teile abzulaufen.

Es wird der Vereinigungsschritt mit der Reihenfolge  $L_i, R_i, L_j, R_j$  behandelt. Mit der Traversierung der Richtungskontur wird am tieferen der beiden Endpunkte  $R_i$  und  $L_j$  begonnen, welcher  $P_{tief}$  genannt wird. Analog dazu wird der andere Endpunkt von  $R_i$  und  $L_j$  mit  $P_{hoch}$  bezeichnet. In Abbildung 4.19(b) ist  $P_{tief}$  der Endpunkt  $R_i$  und  $P_{hoch}$  der Endpunkt  $L_j$ . Beim Ablaufen der Kontur auf dem  $P_{tief}$  wird entweder die Gerade  $g_{P_{hoch, \vec{D}}}$  oder  $g_{P_{hoch, (v_i, v_j)}}$  überquert.

Wenn beim Ablaufen  $g_{P_{hoch, \vec{D}}}$  überquert wird, so wird der bis zu diesem Zeitpunkt abgelaufene Teil verworfen und die beiden Konturen am Schnittpunkt der Projektion zusammengehängt. Wird andernfalls  $g_{P_{hoch, (v_i, v_j)}}$  überquert, so existiert ein Schnittpunkt zwischen beiden Konturen. In diesem Fall wird  $P_{tief}$  zu  $P_{hoch}$  und umgekehrt. An dieser Stelle wird, wie anfangs am neuen Punkt  $P_{tief}$  begonnen und der verdeckte Teil der entsprechenden Kontur abgelaufen bis wieder einer der beiden Fälle auftritt. Wenn eine Kreuzung existiert, dann ist sie gefunden, wenn sich inzidente Segmente von  $P_{tief}$  und  $P_{hoch}$  schneiden. Zuletzt werden die beiden Konturen am Schnittpunkt verschmolzen. Damit ist der Fall  $(L_i, L_j, R_i, R_i)$  abgeschlossen und es werden nur Konturteile abgelaufen, die durch die entsprechende andere Kontur verdeckt sind.

Für Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  mit den entsprechenden Endpunkten und der Reihenfolge  $(L_i, L_j, R_j, R_i)$  ist die Kontur  $C_{v_j, \vec{D}}$  komplett durch die Kontur  $C_{v_i, \vec{D}}$  verdeckt. Somit ist das Resultat der Vereinigung dieser beiden Konturen  $C_{v_i, \vec{D}}$ .

Die Konturen werden im Vereinigungsschritt nach und nach miteinander vereinigt. Da in diesem Schritt die Bewegungsrichtung der zwei Konturen verwendet wird, muss an jedem Teil der Konturen gespeichert sein, welche Richtung das ist. Für zwei Konturen  $C_{v_i, \vec{D}}$  und  $C_{v_j, \vec{D}}$  ist das die Richtung  $(v_i, v_j)$ . Es ist zu beachten, dass die Bestimmung von verdeckten Endpunkten in dieser Beschreibung in konstanter Zeit möglich ist, was für die Laufzeit des Gesamtalgorithmus notwendig ist.

### 4.3.2.3 Bestimmen des Schnitt- oder Projektionspunktes

Da in der Distanzberechnung der Schnittpunkt auf der eingehenden Kante von  $v$  benötigt wird, ist es notwendig, diesen Punkt für jede Richtungskontur im Vereinigungsschritt zu speichern und für den nächsten Rekursionsschritt bereitzustellen. Dazu wird beim paarweisen Vereinigen der Schnittpunkt und Projektionspunkt mit der eingehenden Kante zum Vater gespeichert, wenn vorhanden. Dies kann in konstanter Zeit abgeprüft werden, da dies nur ein neu eingefügtes Segment sein kann, das zwei Konturen verschmilzt oder ein Projektionspunkt in Richtung  $\vec{D}$  ausgehend von den Endpunkten der Richtungskontur.

### 4.3.3 Zeitkomplexität

Im Folgenden wird die Linearzeit für das in Abschnitt 4.3.2 beschriebene Verfahren zum Zeichnen von  $OLP_k$ -Zeichnungen gezeigt. Die zentrale Eigenschaft des rekursiven Verfahrens ist, nur Kontursegmente in der Distanzberechnung und im Vereinigungsschritt abzulaufen, die in den nächsten Rekursionsschritten nicht mehr benötigt werden. Nicht mehr benötigte Segmente von Richtungskonturen sind solche, die nach dem Vereinigungsschritt nicht mehr Teil der Richtungskontur sind und somit wegfallen.

Betrachte den Knoten  $v$  und die Kinder  $v_1, \dots, v_d$  mit den entsprechenden Unterbäumen. Jede Richtungskontur  $C_{v_i, \vec{D}}$  ( $i \in \{1, \dots, d\}$ ) eines Unterbaumes  $T_{v_i}$  in der Zeichnung  $\Gamma(T_{v_i})$  mit der Richtung  $\vec{D} \in \mathcal{D}$  hat zwei Zeiger auf die Endpunkte und einen Zeiger auf den Schnitt- bzw. Projektionspunkt der Kante  $(v, v_i)$ . Die Richtungskonturen sind gespeichert als doppelt verkettete Liste, um eine Traversierung in beide Richtungen zu ermöglichen. Es ist weiter zu beachten, dass die Anzahl der Segmente einer Richtungskontur  $C_{v_i, \vec{D}}$  für  $\vec{D} \in \mathcal{D}$  in  $\mathcal{O}(n)$  ist, siehe Lemma 4.1. Da es für einen Unterbaum gezeichnet auf einem  $k$ -Gitter exakt  $2k$  Richtungskonturen gibt, hat die Kontur  $C_v$  höchstens linear viele Segmente.

**Satz 4.4.** *Das beschriebene Verfahren erstellt Zeichnungen von  $d$ -nären Bäumen mit  $n$  Knoten auf dem  $k$ -Gitter mit  $k \in \{4, 6, 8\}$  und  $d < k$  in Linearzeit.*

*Beweis.* Der Beweis verwendet dieselben Argumente, die für Satz 4.3 verwendet wurden. Da jedem Knoten im gesamten Verfahren nur eine konstante Anzahl von Kontursegmenten zugeteilt wird und in jedem Rekursionsschritt nur Segmente abgelaufen werden, die danach wegfallen, ist die Laufzeit des beschriebenen Verfahrens in  $\mathcal{O}(n)$ .  $\square$

Dasselbe Argument kann auf die Laufzeitberechnung des Algorithmus von Reinhold und Tilford angewendet werden [RT81]. Jeder Konturteil wird darin höchstens zweimal verwendet: beim Erstellen der Segmente und wenn die Distanz berechnet wird. Dabei werden beginnend bei der Wurzel nur Teile abgelaufen, die im Folgenden verworfen werden können.

# 5

## Schluss

---

In dieser Arbeit wurde die Komplexität von Zeichenarten von Bäumen auf verschiedenen Gittern betrachtet. Diese stellen eine Erweiterung des gewöhnlichen 4-er Gitters bzw. orthogonalen Gitters dar. Des Weiteren wurden für die Erweiterungen Zeichenkriterien eingeführt und dazu effiziente Verfahren zum Zeichnen vorgestellt.

### 5.1 Zusammenfassung

Nach einer Einführung wurden zuerst vollständige Bäume auf den erweiterten  $k$ -Gittern mit  $k \in \{4, 6, 8\}$  betrachtet und für diese verschiedene obere und untere Schranken festgestellt. Danach wurde die Komplexität von ordnungserhaltenden und nicht-ordnungserhaltenden Zeichnungen erörtert.

Für ordnungserhaltende Zeichnungen konnte gezeigt werden, dass das Bestimmen der minimalen Fläche auf dem orthogonalen Gitter sowie auf dem 6-er und 8-er Gitter  $\mathcal{NP}$ -hart ist. Durch Hinzunahme der Ästhetikkriterien lokale Uniformität und Pattern hat sich herausgestellt, dass diese Probleme auf dem gewöhnlichen 4-er Gitter  $\mathcal{NP}$ -hart bleiben. Für ordnungserhaltende lokal uniforme Zeichnungen kann dies auf dem 6-er bzw. 8-er Gitter nicht mit dem gleichen Ansatz gezeigt werden. Es konnte jedoch für ordnungserhaltende Pattern- und ordnungserhaltende lokal uniforme Pattern-Zeichnungen ebenfalls gezeigt werden, dass die  $\mathcal{NP}$ -Härte für diese Zeichenart bestehen bleibt. Eine Übersicht der gezeigten Ergebnisse ist in Tabelle 3.4 präsentiert.

Danach wurden Resultate für das Zeichnen mit einheitlichen Kantenlängen für

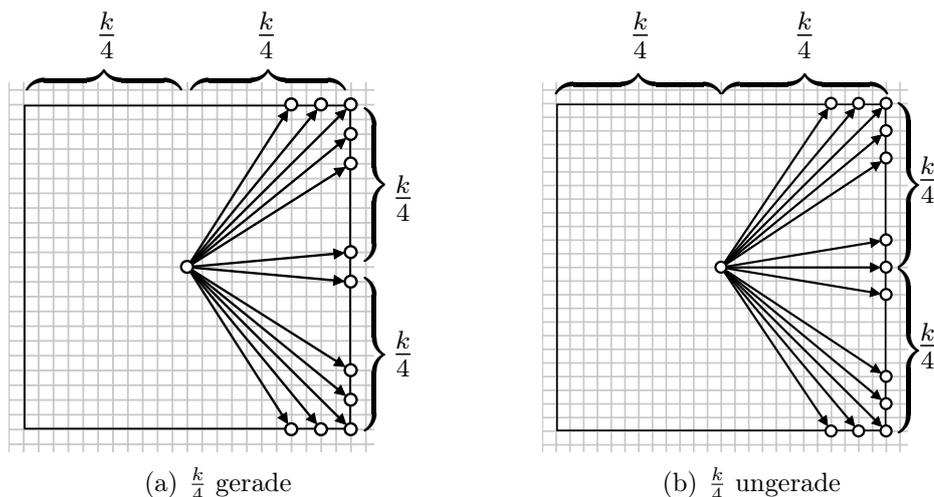
nicht-ordnungserhaltende Zeichnungen festgehalten. Auf dem orthogonalen Gitter wurde bereits gezeigt, dass dies mit vier Richtungen  $\mathcal{NP}$ -hart ist [BC87]. Für die Gitter mit sechs und acht möglichen Richtungen konnte dieses Ergebnis mit Erweiterungen der Reduktionen bestätigt werden. Darüberhinaus wurde für Zeichnungen, die die Ästhetikkriterien lokale Uniformität und Pattern erfüllen, gezeigt, dass es in der Regel  $\mathcal{NP}$ -hart ist, Zeichnungen mit einheitlichen Kantenlängen zu bestimmen. Lediglich auf dem 6-er Gitter ist es unter Beachtung der Pattern nicht gelungen, die  $\mathcal{NP}$ -Härte zu zeigen. Es wird jedoch vermutet, dass diese Probleme ebenfalls  $\mathcal{NP}$ -hart sind, siehe Vermutung 3.2. Eine Zusammenfassung der Resultate für nicht ordnungserhaltende Zeichnungen auf  $k$ -Gittern mit  $k \in \{4, 6, 8\}$  ist in Tabelle 3.5 dargelegt.

Zuletzt wurden in Kapitel 4 Heuristiken zum Zeichnen von ordnungserhaltenden lokal uniformen Pattern-Zeichnungen vorgestellt. Dabei lag einmal das Augenmerk darauf, die Unterbaum Separation in der Zeichnung zu erfüllen und dabei die Fläche der Zeichnungen möglichst klein zu halten, siehe Abschnitt 4.1. Zum Anderen wurde versucht, die Fläche von Zeichnungen ohne Unterbaum Separation zu verringern. Das präsentierte Bottom-Up Verfahren erzeugt Zeichnungen nach einem Ansatz, der bereits für hierarchische Zeichnungen von Bäumen verwendet wurde [RT81]. Ähnlich dazu werden Zeichnungen mit den geforderten Ästhetikkriterien erstellt. Nach einer Beschreibung des Verfahrens in quadratischer Zeit (Abschnitt 4.2) folgt die Beschreibung des laufzeitoptimierten Verfahrens in Abschnitt 4.3, das in Linearzeit implementierbar ist. Dieses wurde für das 6-er Gitter im *Graph Editor Toolkit Gravisto* implementiert [BBF<sup>+</sup>04].

## 5.2 Ausblick

Nach der kurzen Zusammenfassung der Arbeit werden im Folgenden weitere Probleme betrachtet. Zwischen der exponentiellen unteren Schranke für die Fläche von geordneten  $d$ -nären Bäumen mit  $k \in \{6, 8\}$  und  $d < k$  (siehe Tabelle 3.3) und der benötigten Fläche von vollständigen  $d$ -nären Bäumen (siehe Tabelle 3.3) ist noch sehr viel Spielraum. Es stellt sich die Frage, ob eine Klasse von Bäumen existiert, die eine polynomiale untere Schranke für die Fläche besitzen. Auch ist es offen, ob ungeordnete Bäume auf polynomialer Fläche auf das  $k$ -Gitter mit  $k \in \{6, 8\}$  eingebettet werden können.

Nach der Erweiterung des gewöhnlichen orthogonalen Gitters für  $k \in \{6, 8\}$  sind auch Gitter mit noch mehr Richtungen vorstellbar. Die bisher eingeführten  $k$ -Gitter mit  $k \in \{4, 6, 8\}$  können erweitert werden, indem man Gitter für beliebiges durch vier teilbares  $k \in \mathbb{N}$  zulässt. Die Voraussetzung, dass für  $k$  nur Werte erlaubt sind, die durch vier teilbar sind, erfolgt aus Symmetriegründen. Die Verallgemeinerung für  $k \in 4\mathbb{N}$ , wobei  $4\mathbb{N} = \{n \mid n \in \mathbb{N} \wedge n \bmod 4 = 0\}$ , kann wie folgt festgelegt werden: Weiterhin sind im Gitter alle ganzzahligen Koordinaten des  $\mathbb{Z}^2$  mit Knoten besetzt. Jeder Knoten  $v \in V$  mit den Koordinaten  $(x_v, y_v)$  wird mit den Knoten verbunden, die auf den Koordinaten  $(x_v + \frac{k}{4}, y_v + f_{\frac{k}{4}}(t))$ ,  $(x_v - \frac{k}{4}, y_v + f_{\frac{k}{4}}(t) - 2)$  und



**Abbildung 5.1.**  $k$ -Gitter mit  $k$  Richtungen und  $k/2$  Steigungen

$(x_v + f_{\frac{k}{4}}(t), y_v + \frac{k}{4})$ ,  $(x_v - f_{\frac{k}{4}}(t) - 2, y_v - \frac{k}{4})$  liegen. Die Funktion  $f_c(t) : \mathbb{Z} \mapsto \mathbb{Z}$  ist wie folgt definiert:

$$f_c(t) = \begin{cases} 2t, & \text{für gerades } c, & \mathbb{D} = \mathbb{Z} \cap \left[ \lfloor -\frac{c}{2} \rfloor, \lceil \frac{c}{2} \rceil \right] \\ 2t - \text{sgn}(t), & \text{für ungerades } c, & \mathbb{D} = \{ \mathbb{Z} \cap \left[ \lfloor -\frac{c}{2} \rfloor, \lceil \frac{c}{2} \rceil \right] \} \setminus \{0\} \end{cases}$$

In einem  $k$ -Gitter mit  $k \in \{4, 6, 8, 12, 16, \dots\}$  gibt es  $k$  mögliche Kantenrichtungen, da jeder Knoten des  $k$ -Gitters  $k$  inzidente Kanten hat, siehe Abbildung 5.1.

Für diese Erweiterung können die Heuristiken aus Kapitel 4 einfach erweitert und implementiert werden. Es wird vermutet, dass es ebenfalls  $\mathcal{NP}$ -hart ist, die minimale Fläche von ordnungserhaltenden Zeichnungen auf dem  $k$ -Gitter mit  $k \in 4\mathbb{N}$  zu bestimmen. Ebenfalls wird vermutet, dass es  $\mathcal{NP}$ -hart ist, zu bestimmen, ob eine nicht-ordnungserhaltende Zeichnung mit einheitlichen Kantenlängen auf dem  $k$ -Gitter mit  $k \in 4\mathbb{N}$  existiert.

In dieser Arbeit wurden hauptsächlich Bäume auf Gittern untersucht. Für planare Graphen, die auf dem orthogonalen Gitter mit Kantenknicken eingebettet werden, wurde von Tamassia ein Ansatz zum Zeichnen präsentiert [Tam87]. Darin wurde auch die Erweiterung auf das 6-er Gitter erwähnt. Für das 8-er Gitter und für  $k$ -Gitter mit  $k \in 4\mathbb{N}$  ist dies ebenfalls denkbar.



# Abbildungsverzeichnis

1.1	Drei mögliche Zeichnungen Graphen $G$ . . . . .	3
2.1	4-er Gitter . . . . .	9
2.2	Erweiterungen des üblichen orthogonalen Gitters . . . . .	10
2.3	Symmetrisches 6-er Gitter . . . . .	10
2.4	Gitterzeichnungen . . . . .	12
2.5	Spring Embedder (aus [KW01], Abb. 4.1) . . . . .	16
2.6	Azyklischer gerichteter Graph . . . . .	18
3.1	Anwendungsbeispiele für Bäume . . . . .	24
3.2	Erste Ansätze hierarchischer Baumzeichnungen (aus [WS79]) . . . . .	25
3.3	Linksbündig aus [WS79] . . . . .	26
3.4	AE1, AE2 und kleine Breite (aus [WS79]) . . . . .	27
3.5	Rekursionsschritt Reingold und Tilford Algorithmus . . . . .	28
3.6	Beispielzeichnung eines Binärbaumes . . . . .	29
3.7	Untere Schranke in $\Omega(n^2)$ mit Ästhetikkriterien $AK1$ bis $AK4$ . . . . .	30
3.8	Problem der Asymmetrie für $d$ -näre Bäume [Wal90] . . . . .	31
3.9	Hierarchische Baumzeichnung mit rechteckigen Knoten [Blo93] . . . . .	32
3.10	Radialer Ansatz für $d$ -näre Bäume (aus [DBETT99, Ead92]) . . . . .	33
3.11	Radialer Ansatz für $d$ -näre Bäume (aus [DBETT99, Ead92]) . . . . .	34
3.12	Radiale Visualisierung des Gnutella Netzwerks [YF01] . . . . .	35
3.13	Gitterzeichnungen . . . . .	37
3.14	Pattern auf dem 4-er Gitter . . . . .	40
3.15	Pattern auf dem 6-er Gitter . . . . .	40
3.16	Pattern auf dem symmetrischen 6-er Gitter . . . . .	42
3.17	Pattern auf dem 8-er Gitter . . . . .	42
3.18	Beispiele auf dem symmetrischen 6-er Gitter . . . . .	44
3.19	Lokal uniforme Pattern-Zeichnung eines vollständigen 5-nären Baumes . . . . .	46
3.20	Lokal uniforme Pattern-Zeichnung eines vollständigen 4-nären Baumes auf symmetrischem 6-er Gitter . . . . .	46
3.21	$hvd_6$ -Zeichnung eines vollständigen Ternärbaumes . . . . .	47
3.22	Beispielzeichnung eines vollständigen Ternärbaumes mit weniger als $n/9$ Knicken auf dem 6-er Gitter . . . . .	49
3.23	Konstruktionsschritte für den Beweis von Satz 3.4 . . . . .	49

3.24	Vollständiger Ternärbaum mit Blättern außen auf dem 6-er Gitter . . . . .	51
3.25	Leaves-Down Zeichnung eines vollständigen Ternärbaumes auf dem 6-er Gitter . . . . .	52
3.26	Zeichenschema für Satz 3.7 auf dem 8-er Gitter (Gitterkanten weggelassen)	53
3.27	Zeichnung eines vollständigen 7-nären Baumes auf dem 8-er Gitter (Gitterkanten weggelassen) . . . . .	54
3.28	$OLP_8$ -Zeichnung eines vollständigen 5-nären Baumes (Kor. 3.4) . . . . .	55
3.29	Lokal uniforme Zeichnung eines vollständigen Ternärbaumes auf dem 4-er Gitter . . . . .	56
3.30	Untere Schranke für $O_k$ -Zeichnungen für $k \in \{6, 8\}$ . . . . .	58
3.31	Skizze für untere Schranke von $OL_4$ -Zeichnungen . . . . .	60
3.32	Zeichnungen des Variablenbaumes $VT(x_l)$ . . . . .	62
3.33	Literalbaum $LT(y_{i,3})$ mit der Variable $y_{i,3} = x_l$ . . . . .	63
3.34	Literalbaum $LT(y_{i,j})$ von $y_{i,j}$ mit $j = 3$ , $y_{i,j} = \bar{x}_i$ . . . . .	63
3.35	7-närer Baum $T(E)$ auf dem 8-Gitter für den booleschen Ausdruck $E = (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_4)$ und Belegung $\tau(x_1) = \tau(x_2) = false$ und $\tau(x_3) = \tau(x_4) = true$ . . . . .	65
3.36	$O_k$ -Zeichnungen von $T(E)$ für $k \in \{4, 6\}$ . . . . .	69
3.37	$OL_4$ -Zeichnung des Ternärbaum $T(E)$ auf dem 4-er Gitter für den booleschen Ausdruck $E = (\bar{x}_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_4)$ und Belegung $\tau(x_1) = \tau(x_2) = false$ und $\tau(x_3) = \tau(x_4) = true$ . . . . .	71
3.38	Zwei adjazente volle Knoten $u, v$ die nicht lokal uniform gezeichnet werden können . . . . .	72
3.39	Lokal uniforme Zeichnungen durch das Aufsplitten der Kante $(u, v)$ aus Abbildung 3.38 . . . . .	72
3.40	Skizze für Satz 3.17 . . . . .	75
3.41	Ternärisierung . . . . .	76
3.42	Rekursives Winden von Binär- und Ternärbäumen . . . . .	77
3.43	Teilbäume für die Reduktion . . . . .	79
3.44	$T(E)$ von $E = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$ ( $n = 4$ und $r = 3$ ) und Belegung $x_1 = true, x_2 = x_3 = x_4 = false$ . . . . .	82
3.45	Voller 6-er Baum und 6-er Kodierungsbaum . . . . .	86
3.46	$T(E)$ von $E = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$ ( $n = 4$ und $r = 3$ ) und Belegung $x_1 = true, x_2 = x_3 = x_4 = false$ auf dem 6-er Gitter	88
3.47	$T(E)$ von $E = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee x_3 \vee \bar{x}_4)$ ( $n = 4$ und $r = 3$ ) und Belegung $x_1 = true, x_2 = x_3 = x_4 = false$ . . . . .	94
4.1	Ausschnitt eines Baumes auf dem 6-erGitter mit Knoten $v$ und den Kindern $v_i$ und $v_j$ . . . . .	97
4.2	Relative Bewegungsrichtung von $v_j$ in Abhängigkeit von $v_i$ und der entsprechende Verkürzungsfaktor $ \vec{D} $ . . . . .	98
4.3	Relative Bewegungsrichtungen auf dem 8-er Gitter . . . . .	99
4.4	Bewegungsrichtungen auf dem symmetrischen und asymmetrischen 6-er Gitter . . . . .	100

4.5	Bewegungsrichtungen auf dem 4-er Gitter . . . . .	101
4.6	$\vec{D}$ -maximale Knoten und konvexe Kontur einer Zeichnung $\Gamma(G)$ auf dem 8-er Gitter . . . . .	103
4.7	Abstandsberechnungen zweier Unterbäume $T_{v_i}$ und $T_{v_j}$ auf dem symmetrischen 6-er Gitter . . . . .	105
4.8	Verkürzung der Kanten $(v, v_i)$ und $(v, v_j)$ aus Abbildung 4.7 . . . . .	109
4.9	Negativbeispiel von Algorithmus 4.2 auf dem symmetrischen 6-er Gitter . . . . .	110
4.10	Richtungskonturen für die $D_1, D_2, D_3 \in \mathcal{D}_8$ . . . . .	112
4.11	Kontur $C_{\vec{D}}$ eines Pfades für die Richtung $\vec{D} = (-1, 1)$ . . . . .	113
4.12	Distanzberechnung mit exakten Konturen zwischen $C_{v_i, \vec{D}}$ und $C_{v_j, -\vec{D}}$ auf dem 6-er Gitter . . . . .	114
4.13	Beispiel zu den Berechnungen im Rekursionsschritt mit Kindunterbäumen $T(v_i), T(v_j)$ und $T(v_k)$ von $v$ . . . . .	115
4.14	Negativbeispiel Algorithmus 4.2 . . . . .	119
4.15	Erweitern der Konturen in der Distanzberechnung . . . . .	121
4.16	Fälle im Vereinigungsschritt für die Reihenfolge $(L_i, L_j, R_i, R_j)$ der Projektion in Richtung $\vec{D} = (0, 1)$ . . . . .	123
4.17	Fälle im Vereinigungsschritt für die Reihenfolge $(L_i, L_j, R_j, R_i)$ der Projektion in Richtung $\vec{D} = (0, 1)$ . . . . .	123
4.18	Verkürzen der ausgehenden Kanten und resultierende Konturen $\vec{D} = (0, 1)$ und $\vec{D}^o = (1, 0)$ . . . . .	125
4.19	Vereinigung zweier Konturen mit teilweiser Verdeckung $(L_i, L_j, R_i, R_j)$ nach dem Verkürzungsschritt . . . . .	127
5.1	$k$ -Gitter mit $k$ Richtungen und $k/2$ Steigungen . . . . .	133



# Tabellenverzeichnis

3.1	Flächenschränken für Binärbäume . . . . .	36
3.2	Flächenschränken für vollständige $d$ -näre Bäume auf dem $k$ -Gitter . . . . .	45
3.3	Untere Schranken für die Fläche von ordnungserhaltende Zeichnungen auf $k$ -Gittern mit $k \in \{4, 6, 8\}$ . . . . .	58
3.4	Minimale Fläche für ordnungserhaltende Zeichnungen . . . . .	61
3.5	Einheitliche Kantenlänge nicht ordnungserhaltender Zeichnungen $\mathcal{NP}$ -hart	78



# Definitionsverzeichnis

2.1	Graph	5
2.2	Einbettung, Zeichnung	7
2.3	Zeichenkonvention	8
2.4	Gitter, Gittergraph	9
2.5	Gittereinbettung	11
2.6	Distanz, Abstand	11
2.7	Ästhetikkriterium	11
3.1	Baum, geordnet, ungeordnet, Wurzelbaum	21
3.2	<i>AK1</i>	23
3.3	<i>AK2</i>	26
3.4	<i>AK3</i>	26
3.5	Isomorphie	27
3.6	<i>AK4</i>	27
3.7	Pattern-Zeichnung	41
3.8	Global uniforme Zeichnung	43
3.9	Lokal uniforme Zeichnung	43
3.10	3-SAT	61
3.11	NOT-ALL-EQUAL-3-SAT, <i>NAE3SAT</i>	78
4.1		99
4.2	Richtungsvektor	101
4.3	Richtungsabstand	102
4.4	$\vec{D}$ -maximaler Knoten, $\vec{D}$ -maximale Gerade	102
4.5	Konvexe Kontur	103
4.6	Richtungskontur, Kontur	112



# Algorithmenverzeichnis

4.1	kompaktiereZeichnungMitSubtreeSeparation . . . . .	104
4.2	kompaktiereZeichnungMitUnterbaumSeparation . . . . .	107
4.3	kompaktiereZeichnungOhneUnterbaumSeparation . . . . .	116



# Literaturverzeichnis

- [ABJT04] T. Akkerman, C. Buchheim, M. Jünger und D. Teske. On the complexity of drawing trees nicely: Corrigendum. *Acta Inf.*, 40(8):603–607, 2004.
- [AHU74] A. V. Aho, J. E. Hopcroft und J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [BBF<sup>+</sup>04] C. Bachmaier, F. J. Brandenburg, M. Forster, P. Holleis und M. Raitner. Gravisto: Graph visualization toolkit. In *Graph Drawing*, Seiten 502–503, 2004.
- [BBS05] C. Bachmaier, U. Brandes und B. Schlieper. Drawing phylogenetic trees. In *ISAAC*, Volume 3827 von *Lecture Notes in Computer Science*, Seiten 1110–1121. Springer, 2005.
- [BC87] S. N. Bhatt und S. Cosmadakis. The complexity of minimizing wire lengths in VLSI layouts. *Inf. Process. Lett.*, 25(4):263–267, 1987.
- [BJL02] C. Buchheim, M. Jünger und S. Leipert. Improving walker’s algorithm to run in linear time. In S. G. Kobourov und M. T. Goodrich, editors, *Graph Drawing*, Volume 2528 von *Lecture Notes in Computer Science*, Seiten 344–353. Springer, 2002.
- [BK80] R. P. Brent und H. T. Kung. On the area of binary tree layouts. *Inf. Process. Lett.*, 11(1):46–48, 1980.
- [BK01] U. Brandes und B. Köpf. Fast and simple horizontal coordinate assignment. In Petra Mutzel, Michael Jünger und Sebastian Leipert, editors, *Proc. Graph Drawing, GD 2001*, Volume 2265 von *LNCS*, Seiten 31–44. Springer, 2001.
- [BL76] S. Booth, K und G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13(3):335–379, December 1976.
- [BL83] S. N. Bhatt und C. E. Leiserson. *Minimizing the longest edge in a VLSI Layout*. MIT Industrial Liaison Program report. MIT, 1983.

- [Blo93] A. Bloesch. Aesthetic layout of generalized trees. *Softw. Pract. Exper.*, 23(8):817–827, 1993.
- [Bra90] F. J. Brandenburg. Nice drawings of graphs are computationally hard. In *Selected contributions of the seventh interdisciplinary workshop on informatics and psychology on Visualization in human-computer interaction*, Seiten 1–15, New York, NY, USA, 1990. Springer-Verlag New York, Inc.
- [CDBP92] P. Crescenzi, G. Di Battista und A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Comput. Geom. Theory Appl.*, 2:187–200, 1992.
- [CGKT97] T. Chan, M. T. Goodrich, S. R. Kosaraju und R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. In *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, Seiten 63–75. Springer-Verlag, 1997.
- [CGKT02] T. M. Chan, M. T. Goodrich, S. R. Kosaraju und R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. *Comput. Geom. Theory Appl.*, 23(2):153–162, 2002.
- [Cha99] T. M. Chan. A near-linear area bound for drawing binary trees. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '99, Seiten 161–168, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.
- [CNAO85] N. Chiba, T. Nishizeki, S. Abe und T. Ozawa. A linear algorithm for embedding planar graphs using pq-trees. *J. Comput. Syst. Sci.*, 30:54–76, 1985.
- [CON85] N. Chiba, K. Onoguchi und T. Nishizeki. Drawing plane graphs nicely. *Acta Inf.*, 22(2):187–201, 1985.
- [CP95] M. Chrobak und T. Payne. A linear-time algorithm for drawing planar graphs. *Information Processing Letters*, 1995.
- [CP98] P. Crescenzi und P. Penna. Strictly-upward drawings of ordered search trees. *Theor. Comput. Sci.*, 203(1):51–67, 1998.
- [DBETT94] G. Di Battista, P. Eades, R. Tamassia und I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4:235–282, 1994.
- [DBETT99] G. Di Battista, P. Eades, R. Tamassia und I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

- [DFPP90] H. De Fraysseix, J. Pach und R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.
- [DH96] R. Davidson und D. Harel. Drawing graphs nicely using simulated annealing. *ACM Trans. Graph.*, 15:301–331, October 1996.
- [Ead84] P. Eades. A heuristic for graph drawing. In *Congressus Numerantium*, Volume 42, Seiten 149–160, 1984.
- [Ead92] P. Eades. Drawing free trees. *Bull. Inst. Combin. Appl.*, 5:10–36, 1992.
- [EK86] P. Eades und D. Kelly. Heuristics for reducing crossings in 2-layered networks. *Ars Combinatoria*, 21(A):89–98, 1986.
- [ELL92] P. Eades, T. Lin und X. Lin. Minimum size h-v drawings. In *Advanced Visual Interfaces*, Seiten 386–394, 1992.
- [ELL93] P. Eades, T. Lin und X. Lin. Two tree drawing conventions. *Internat. J. Comput. Geom. Appl.*, 3(2):133–153, 1993.
- [Fá48] I. Fáry. On straight line representations of planar graphs. *Acta Sci. Math.*, 11:229–233, 1948.
- [FR91] T. M. J. Fruchterman und E. M. Reingold. Graph drawing by force-directed placement. *Softw: Pract. Exper.*, 21(11):1129–1164, November 1991.
- [Fra07] F. Frati. Straight-line orthogonal drawings of binary and ternary trees. In Seok-Hee Hong, Takao Nishizeki und Wu Quan, editors, *GD 2007*, Volume 4875 von *LNCS*, Seiten 76–87, Heidelberg, 2007. Springer.
- [GGT93] A. Garg, M. T. Goodrich und R. Tamassia. Area-efficient upward tree drawings. In *Proceedings of the ninth annual symposium on Computational geometry*, SCG '93, Seiten 359–368, New York, NY, USA, 1993. ACM.
- [GGT96] A. Garg, M. T. Goodrich und R. Tamassia. Planar upward tree drawings with optimal area. *Int. J. Comput. Geometry Appl.*, 6(3):333–356, 1996.
- [GJ90] M. R. Garey und D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, USA, 1990.
- [Gre89] A. Gregori. Unit-length embedding of binary trees on a square grid. *Inf. Process. Lett.*, 31(4):167 – 173, 1989.
- [Her89] J. Hershberger. Finding the upper envelope of  $n$  line segments in  $O(n \log n)$  time. *Inf. Process. Lett.*, 33:169–174, December 1989.

- [HT74] J. Hopcroft und R. Tarjan. Efficient planarity testing. *J. ACM*, 21:549–568, October 1974.
- [JG72] E. G. Coffman Jr. und R. L. Graham. Optimal scheduling for two processor systems. *Acta Inf.*, 1(3):200–213, 1972.
- [Kan92] G. Kant. Hexagonal grid drawings. In Ernst W. Mayr, Hrsg., *WG 1992*, Volume 675 von *LNCS*, Seiten 263–276, Heidelberg, 1992. Springer.
- [Kim95] S. K. Kim. Simple algorithms for orthogonal upward drawings of binary and ternary trees. In *7th Canadian Conference on Computational Geometry*, Seiten 115–120, 1995.
- [KK89] T. Kamada und S. Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31:7–15, April 1989.
- [KMS94] C. Kosak, J. Marks und S. M. Shieber. Automating the layout of network diagrams with specified visual organization. *IEEE Trans. Syst., Man, Cybern.*, 24(3):440–454, 1994.
- [Knu68] D. E. Knuth. *The Art of Computer Programming*, Volume 1. Addison-Wesley, 1968.
- [Knu71] D. E. Knuth. Optimum binary search trees. *Act. Inf.*, 1:14–25, 1971.
- [Kru64] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, March 1964.
- [Kur30] K. Kuratowski. Sur le Probleme des Courbes Gauches en Topologie. *Fundamenta Mathematicae*, 15:271–283, 1930.
- [KW01] M. Kaufmann und D. Wagner. *Drawing Graphs*, Volume 2025 von *LNCS*. Springer, Heidelberg, 2001.
- [LEC67] A. Lempel, S. Even und I. Cederbaum. An algorithm for planarity testing of graphs. In Pierre Rosenstiehl, Hrsg., *Theory of Graphs, International Symposium, Rome*, Seiten 215–232. Gordon and Breach, 1967.
- [Lei80] C. E. Leiserson. Area-efficient graph layouts. In *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, Seiten 270–281, Washington, DC, USA, 1980. IEEE Computer Society.
- [Lei84] F. T. Leighton. New lower bound techniques for VLSI. *Theory Comput. Syst.*, 17:47–70, 1984.
- [LNS85] R. J. Lipton, S. C. North und J. S. Sandberg. A method for drawing graphs. In *Proceedings of the first annual symposium on Computational geometry*, SCG '85, Seiten 153–160, New York, NY, USA, 1985. ACM.

- [MS04] K. Marriott und P. J. Stuckey. NP-completeness of minimal width unordered tree layout. *J. Graph Algorithms Appl.*, 8(2):295–312, 2004.
- [PRS81] M. S. Paterson, W. L. Ruzzo und L. Snyder. Bounds on minimax edge length for complete binary trees. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, STOC '81, Seiten 293–299, New York, NY, USA, 1981. ACM.
- [RS81] W. L. Ruzzo und L. Snyder. Minimum edge length planar embeddings of trees VLSI systems and computation. *VLSI Systems and Computations*, Seiten 119–123, 1981.
- [RT81] E. M. Reingold und J. S. Tilford. Tidier drawing of trees. *IEEE Trans. Software Eng.*, 7(2):223–228, 1981.
- [Sch90] W. Schnyder. Embedding planar graphs on the grid. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, SODA '90, Seiten 138–148, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.
- [She62] R. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2):125–140, June 1962.
- [Shi76] Y. Shiloach. *Arrangements of Planar Graphs on the Planar Lattice*. Doktorarbeit, Dep. Appl. Math., Weizmann Instit. Sci., Rehovot, Israel, 1976.
- [SKC00] C. Shin, S. K. Kim und K.-Y. Chwa. Area-efficient algorithms for straight-line tree drawings. *Comput. Geom. Theory Appl.*, 15(4):175–202, 2000.
- [SKKC98] C. Shin, S. K. Kim, S. Kim und K.-Y. Chwa. Algorithms for drawing binary trees in the plane. *Inf. Process. Lett.*, 66(3):133 – 139, 1998.
- [SM95] K. Sugiyama und K. Misue. Graph drawing by the magnetic spring model. *J. Vis. Lang. Comput.*, 6(3):217–231, 1995.
- [SR82] K. J. Supowit und E. M. Reingold. The complexity of drawing trees nicely. *Acta Inf.*, 18:377–392, 1982.
- [STT81] K. Sugiyama, S. Tagawa und M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst., Man, and Cyber.*, 11(2):109–125, 1981.
- [Tam87] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.

- [TDBB88] R. Tamassia, G. Di Battista und C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybern.*, 18:61–79, 1988.
- [Tre96] L. Trevisan. A note on minimum-area upward drawing of complete and fibonacci trees. *Inf. Process. Lett.*, 57:231–236, 1996.
- [Tut63] W. T. Tutte. How to draw a graph. *Proc. Lond. Math. Soc.*, 13:743–767, 1963.
- [Ull75] J. D. Ullman. NP-complete scheduling problems. *Journal of Computer and System Sciences*, 10:384–393, 1975.
- [Val81] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Computers*, 30(2):135–140, 1981.
- [Wal90] J. Q. Walker. A node-positioning algorithm for general trees. *Softw. Pract. Exper.*, 20(7):685–705, 1990.
- [WM95] X. Wang und I. Miyamoto. Generating customized layouts. In *Graph Drawing*, Lecture Notes in Computer Science, Seiten 504–515. Springer, 1995.
- [Woo81] D. R. Woods. *Drawing planar graphs*. Ph. D. Stanford University, Computer Science Dept., 1981.
- [WS79] C. S. Wetherell und A. Shannon. Tidy drawings of trees. *IEEE Trans. Softw. Eng.*, 5(5):514–520, 1979.
- [YF01] K.-P. Yee und D. Fisher. Animated exploration of dynamic graphs with radial layout. In *INFOVIS*, Seiten 43–50, 2001.