

# Geordnete Spann bäume

Sven Wagner

sven.wagner@uni-konstanz.de

## Zusammenfassung

Diese Seminararbeit befasst sich mit einer Verallgemeinerung der kanonischen Ordnungen für nicht notwendig dreifach zusammenhängende Graphen: den geordneten Spann bäumen. Dies sind Spann bäume, deren Knoten alle eine gewisse Ordnungseigenschaft erfüllen. Es wird ein Algorithmus vorgestellt, der zu jedem zusammenhängenden planaren Graphen in Linearzeit eine planare Einbettung mit zugehörigem geordnetem Spannbaum berechnet. Zudem wird eine Anwendung dieses algorithmischen Werkzeuges für das Zeichnen von Graphen, genauer für das Zeichnen von 2-Sichtbarkeitsdarstellungen triangulierter Graphen, aufgezeigt.

## 1 Einleitung

Der hier vorliegende Text ist eine Ausarbeitung eines Vortrages im Seminar *Zeichnen von Graphen* von Prof. Dr. U. Brandes an der Universität Konstanz im Wintersemester 2004/05 ([5]). Die Vorträge in diesem Seminar behandelten aktuelle Verfahren und Algorithmen, die für das Zeichnen von Graphen von Bedeutung sind.

Mein Vortrag basierte auf dem Artikel *Orderly Spanning Trees with Applications to Graph Encoding and Graph Drawing* von Chiang, Lin und Lu, erschienen in *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms* ([1]). Dabei habe ich eine überarbeitete und erweiterte Version des Artikels verwendet (siehe ebenfalls [1]).

Nach der Erläuterung der erforderlichen Grundlagen in Abschnitt 2, beschäftigt sich diese Ausarbeitung in Abschnitt 3 mit den *geordneten Spann bäumen* und ihrer Beziehung zu den schon früher bekannten kanonischen Ordnungen, die nur für dreifach zusammenhängende planare Graphen existieren.

Allerdings gibt es auch nicht zu jeder planaren Einbettung eines zusammenhängenden Graphen einen geordneten Spannbaum. Es wird daher in Abschnitt 4 ein Algorithmus vorgestellt, der aus einem gegebenen zusammenhängenden planaren Graphen in Linearzeit, sowohl eine geeignete planare Einbettung, als auch einen zugehörigen geordneten Spannbaum berechnet.

Im letzten Abschnitt wird eine Anwendung der geordneten Spann bäume aufgezeigt: Ein Linearzeitalgorithmus zum Zeichnen von platzoptimalen 2-Sichtbarkeitsdarstellungen triangulierter Graphen.

## 2 Grundbegriffe

In dieser Ausarbeitung betrachten wir stets einfache planare ungerichtete Graphen. Dabei heißt ein Graph einfach, wenn es zwischen je zwei Knoten in diesem Graphen höchstens eine Kante gibt. Er heißt planar, wenn eine planare Einbettung, also eine kreuzungsfreie Zeichnung in der Ebene, existiert.

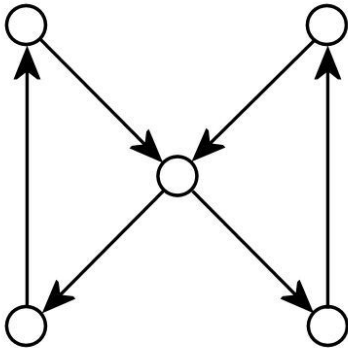
Insbesondere werden wir Graphen mit einer der folgenden zusätzlichen Eigenschaften betrachten:

- **zusammenhängend:** In diesem Graphen sind je zwei Knoten gegenseitig erreichbar, d.h. es existiert zwischen je zwei Knoten ein Pfad.
- **zweifach zusammenhängend:** Der Graph ist zusammenhängend und es gilt: Entfernt man einen Knoten (und damit auch alle zu ihm inzidenten Kanten) aus dem Graphen, so ist der Graph immernoch zusammenhängend.

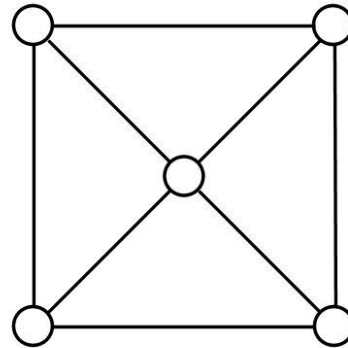
- **n-fach zusammenhängend:** Der Graph ist zusammenhängend und es gilt: Entfernt man  $n - 1$  Knoten (und damit auch alle zu ihnen inzidenten Kanten) aus dem Graphen, so ist der Graph immernoch zusammenhängend.
- **trianguliert:** Jede Facette (das sind die durch die Knoten und Kanten begrenzten maximalen zusammenhängenden Teilmengen der Ebene) des Graphen ist zu genau drei Knoten und drei Kanten inzident. Triangulierte Graphen sind insbesondere dreifach zusammenhängend.

## 2.1 Beispiel:

Die Abbildung 2.1 zeigt einen zusammenhängenden Graphen. Dabei liegen alle Knoten auf dem eingezeichneten Kreis. Da aber z.B. von einem Knoten auf der rechten Seite jeder Pfad, der zu einem Knoten auf der linken Seite führt, über den mittleren Knoten verlaufen muß, ist der Graph nicht zweifach zusammenhängend. Einen zweifach zusammenhängenden Graphen zeigt die Abbildung 2.2.

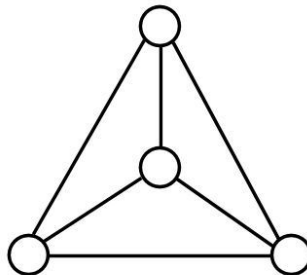


2.1



2.2

Einen triangulierten Graphen sehen wir in der Abbildung 2.3.



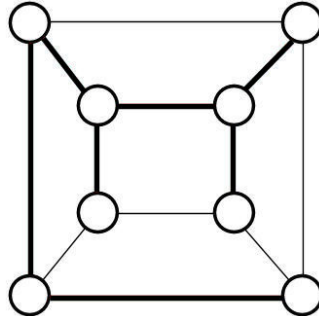
2.3

Bei der Betrachtung von zusammenhängenden Graphen werden in dieser Ausarbeitung außerdem folgende Begriffe verwendet:

- **Rand/Kontur:** Der Rand eines Graphen besteht aus den Knoten und Kanten, die zu der äußeren Facette des Graphen inzident sind.
- **externe/interne Knoten/Kanten:** Die Knoten bzw. Kanten auf dem Rand des Graphen heißen extern, alle anderen (also die, die im Inneren des Graphen liegen) heißen intern.
- **Spannbaum:** Ein Spannbaum eines Graphen ist ein Teilgraph, der alle Knoten des Graphen enthält und ein Baum ist.

## 2.2 Beispiel:

Die Abbildung 2.4 zeigt einen zusammenhängenden Graphen mit eingezeichnetem Spannbaum.



2.4

Haben wir nun einen Spannbaum  $T$  eines zusammenhängenden planar eingebetteten Graphen gegeben, so sind für uns folgende Definitionen von Bedeutung:

- **Präordnung im Gegenuhrzeigersinn:** Diese Präordnung der Knoten in dem Baum erhält man, indem man die Knoten des Baumes mit folgendem Algorithmus durchläuft:  $CWP(\text{Wurzel}(T))$ .

$CWP(v)$  ( $v \in T$ )

Falls  $v$  kein Blatt von  $T$  ist, führe aus:

Durchlaufe die Kinder  $w$  von  $v$  in  $T$  im Gegenuhrzeigersinn um  $v$  (also von links nach rechts) und führe dabei für jedes dieser Kinder aus:

$CWP(w)$

- **verwandte Knoten:** Zwei Knoten in  $T$  heißen verwandt (bzgl.  $T$ ), falls einer der Knoten ein Nachfolger des anderen im Baum  $T$  ist.
- **geordnete Knoten:** Ein Knoten  $v$  in  $T$  heißt geordnet, falls die Kanten zu seinen Nachbarn im Gegenuhrzeigersinn um  $v$  folgende (eventuell leeren) Blöcke bilden:

B1( $v$ ) Die Baumkante zu seinem direkten Vorgänger (Elternknoten) in  $T$ .

B2( $v$ ) Die Nichtbaumkanten zu allen nichtverwandten Nachbarn von  $v$  im Graphen, die bezüglich der Präordnung im Gegenuhrzeigersinn von  $T$  vor  $v$  liegen.

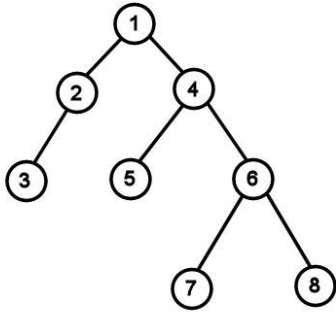
B3( $v$ ) Die Baumkanten zu seinen direkten Nachfolgern (Kindern) in  $T$ .

B4( $v$ ) Die Nichtbaumkanten zu allen nichtverwandten Nachbarn von  $v$  im Graphen, die bezüglich der Präordnung von  $T$  hinter  $v$  liegen.

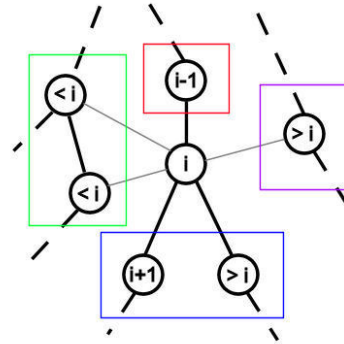
Inbesondere hat  $v$  außer seinem Elternknoten und seinen Kindern keine verwandten Nachbarn.

## 2.3 Beispiel:

Die Abbildung 2.5 zeigt einen Baum mit der Präordnung im Gegenuhrzeigersinn. Abbildung 2.6 zeigt einen geordneten Knoten mit der zugehörigen Einteilung der Nachbarn in die Blöcke  $B_1$  bis  $B_4$ .



2.5



2.6

### 3 Geordnete Spannbäume und kanonische Ordnungen

In diesem Abschnitt definieren wir die geordneten Spannbäume und betrachten die Beziehung zu den kanonischen Ordnungen.

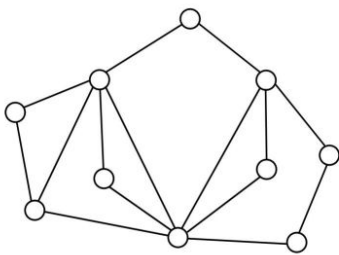
#### 3.1 Definition:

Sei  $G$  ein zusammenhängender planarer Graph und sei  $H$  eine planare Einbettung von  $G$ . Dann heißt ein Spannbaum  $T$  von  $H$  **geordneter Spannbaum** von  $H$ , falls alle seine Knoten geordnet sind und seine Wurzel auf dem Rand von  $H$  liegt.

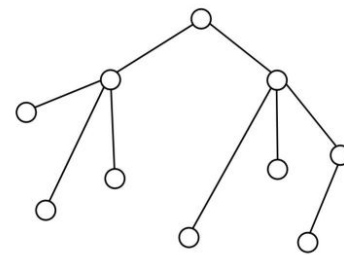
Ein Paar  $(H, T)$ , wobei  $H$  eine planare Einbettung von  $G$  und  $T$  ein geordneter Spannbaum (mit Wurzel  $r$ ) von  $H$  ist, heißt ein **geordnetes Paar** (bzgl.  $r$ ) von  $G$ .

#### 3.2 Beispiel:

Die Abbildungen 3.1 und 3.2 zeigen einen zusammenhängenden planar eingebetteten Graphen und einen zugehörigen geordneten Spannbaum.



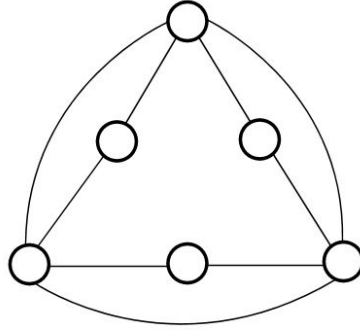
3.1



3.2

Wie wir später sehen werden können wir zu jedem planaren zusammenhängenden Graphen eine planare Einbettung und einen zugehörigen geordneten Spannbaum berechnen, allerdings existiert nicht zu jeder planaren Einbettung ein geordneter Spannbaum:

Abbildung 3.3 zeigt einen zusammenhängenden planar eingebetteten Graphen, zu dem kein geordneter Spannbaum existiert.



3.3

Haben wir ein solches Paar  $(H, T)$  gegeben, erhalten wir durch die Präordnung im Gegenuhrzeigersinn von  $T$  eine Ordnung der Knoten von  $H$ . Diese ist für triangulierte Graphen eine sogenannte *kanonische Ordnung*. Mehr noch: Haben wir einen für einen dreifach zusammenhängenden planar eingebetteten Graphen eine kanonische Ordnung gegeben, so induziert diese immer einen geordneten Spannbaum auf diesem Graphen. Somit stellen die geordneten Spannbäume eine Verallgemeinerung der kanonischen Ordnung für beliebige zusammenhängende Graphen dar.

Für die Definition der kanonischen Ordnung benötigen wir erst einmal die folgenden Bezeichnungen:

### 3.3 Definition:

Sei  $H$  ein (dreifach zusammenhängender) planar eingebetteter Graph. Sei  $v_1, \dots, v_n$  eine Ordnung der Knoten von  $H$ . Sei  $H_i$  der von den Knoten  $v_1, \dots, v_i$  aufgespannte Teilgraph von  $H$ . Ist  $H_2$  zusammenhängend (d.h. er besteht aus den Knoten  $v_1, v_2$  und der Kante  $\{v_1, v_2\}$ ), so nennen wir  $v_1$  den **linken** Knoten und  $v_2$  den **rechten** Knoten von  $H_2$ .

Ist  $H(i)$  für  $i > 2$  zweifach zusammenhängend, so gibt es einen Pfad  $(v_1 =: u_1, u_2, \dots, u_l =: v_2)$  auf dem Rand von  $H_i$ , der nicht über die Kante  $\{v_1, v_2\}$  führt. Seien  $x = u_{k_1}, y = u_{k_2}$  mit  $x \neq y$  zwei Knoten auf dem Rand von  $H(i)$ . Dann sagen wir "x liegt weiter **links** als y", falls  $k_1 < k_2$  gilt.

Also ist immer  $v_1$  der am weitesten links und  $v_2$  der am weitesten rechts gelegene Knoten.

Nun können wir die kanonischen Ordnungen definieren:

### 3.4 Definition:

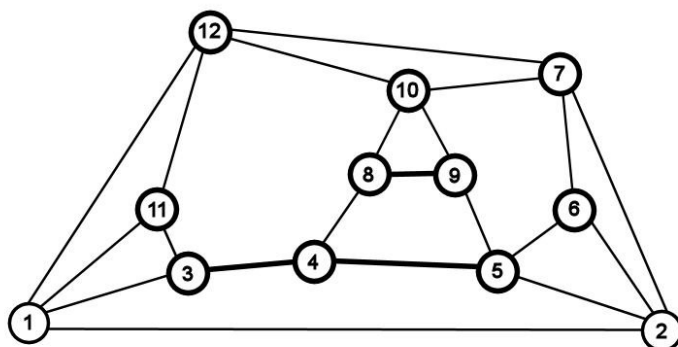
Sei  $v_1, \dots, v_n$  eine Ordnung der Knoten eines dreifach zusammenhängenden planar eingebetteten Graphen  $H$ . Diese Ordnung heißt **kanonische Ordnung**, falls die Kante  $\{v_1, v_2\}$  auf dem Rand von  $H$  liegt, und geordnete Intervalle  $I_1, \dots, I_K$  existieren, die das Intervall  $[3, n]$  so partitionieren, daß folgende Eigenschaften für alle  $1 \leq j \leq K$  gelten:

Sei  $I_j = [k, k + q]$  und  $C_j$  der Pfad  $(v_k, \dots, v_{k+q})$ .

- Der Graph  $H_{k+q}$  ist zweifach zusammenhängend. Sein Rand enthält die Kante  $\{v_1, v_2\}$  und den Pfad  $C_j$ .  $C_j$  hat keine Sehnen in  $H$  (d.h. es gibt in  $H$  keine Kante  $\{v_{i_1}, v_{i_2}\}$  mit  $k \leq i_1 < i_1 + 1 < i_2 \leq k + q$ ).
- Falls  $q = 0$  ist, so hat  $v_k$  mindestens zwei Nachbarn in  $H_{k-1}$ , alle liegen auf dem Rand dieses Teilgraphen.
- Falls  $q > 0$  ist, so hat  $C_j$  genau zwei Nachbarn auf dem Rand von  $H_{k-1}$ . Der weiter links gelegene Nachbar ist nur zu  $v_k$  inzident, der weiter rechts gelegene Nachbar nur zu  $v_{k+q}$ .
- Jeder Knoten  $v_i$  mit  $k \leq i \leq k + q$  und  $i < n$  hat mindestens einen Nachbarn in  $H \setminus H_{k+q}$ .

Zur Veranschaulichung dieser Definition betrachten wir das folgende

### 3.5 Beispiel:

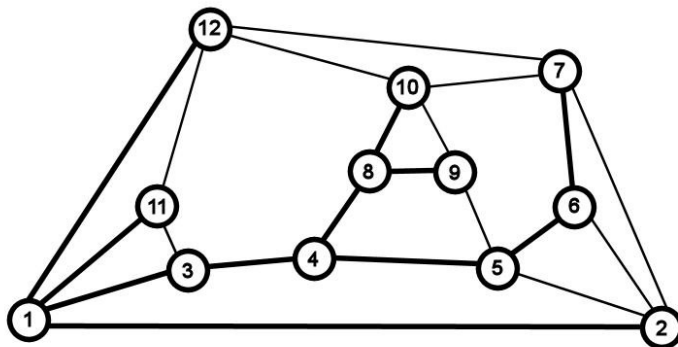


3.4

Abbildung 3.4 zeigt einen dreifach zusammenhängenden planar eingebetteten Graphen  $H$  und eine kanonische Ordnung von  $H$ . Die fettgedruckten Knoten und Kanten zeigen die Intervalle der kanonischen Ordnung:

| $j$ | $I_j$ |
|-----|-------|
| 1   | 3,4,5 |
| 2   | 6     |
| 3   | 7     |
| 4   | 8,9   |
| 5   | 10    |
| 6   | 11    |
| 7   | 12    |

Verbinden wir nun die Pfade  $C_j$  mit ihren am weitesten links gelegenen Nachbarn, so erhalten wir zusammen mit der Kante  $\{v_1, v_2\}$  einen Spannbaum des Graphen. Dieser ist sogar geordnet, wie man leicht überprüfen kann.



3.5

Diese Konstruktion kann man für einen beliebigen dreifach zusammenhängenden planar eingebetteten Graphen  $H$  mit einer kanonischen Ordnung  $v_1, \dots, v_n$  durchführen:

### 3.6 Definition:

Sei  $T$  die Kante  $\{v_1, v_2\}$  zusammen mit der Vereinigung der Pfade  $(v_l, v_k, \dots, v_{k+q})$  über alle Intervalle  $I_j = [k, k+q]$  ( $1 \leq j \leq K$ ), wobei  $v_l$  der am weitesten links gelegene Nachbar von  $v_k$  auf dem Rand von  $H_{k-1}$  ist.

Dann ist  $T$  ein Spannbaum von  $H$  mit Wurzel  $v_1$ , genannt ein **kanonischer Spannbaum** von  $H$ .

Wie im Beispiel 3.5 erhalten wir allgemein das folgende Ergebnis:

### 3.7 Satz:

Sei  $T$  ein kanonischer Spannbaum eines dreifach zusammenhängenden planar eingebetteten Graphen  $H$ . Dann ist  $T$  ein geordneter Spannbaum von  $H$ .

Beweis:

Sei  $v_1, \dots, v_n$  die Präordnung im Gegenuhrzeigersinn der Knoten von  $T$ .

Zuerst zeigen wir:

Ist  $\{v, w\}$  eine Nichtbaumkante, so sind  $v$  und  $w$  nicht verwandt.

Nehmen wir an,  $v$  und  $w$  wären verwandt.

Dazu sei ohne Einschränkung der Allgemeinheit  $v = v_i$  und  $w = v_j$  mit  $i < j$ .

Liegt  $w$  in dem Intervall  $I$  der kanonischen Ordnung, so ist  $v$  nicht in  $I$ , da der zu  $I$  gehörige Pfad  $C_I$  keine Sehnen in  $H$  hat.

Wir unterscheiden nun die folgenden Fälle:

1. Fall:  $I = [j, j]$ .

Dann wäre  $v_i$  weiter links gelegen als der Elternknoten von  $v_j$ , da beide auf dem Rand von  $G_{j-1}$  liegen müssen und  $v_i$  ein Vorgänger des Elternknotens von  $v_j$  ist, also näher bei  $v_1$  liegt.  $\zeta$  zur Konstruktion des kanonischen Spannbaumes  $T$ .

2. Fall:  $I = [k, k + q]$  mit  $q > 0$ .

Wegen  $v_i \in G_{k-1}$ , folgt  $v_j = v_k$  oder  $v_j = v_{k+q}$ .

2.1. Fall:  $v_j = v_k$ .

Dann hätte  $v_k$  zwei Nachbarn in  $G_{k-1}$ .  $\zeta$  zu  $v_j$  hat genau einen Nachbarn in  $G_{k-1}$  (Definition kanonische Ordnung).

2.2. Fall:  $v_j = v_{k+q}$ .

Dann wäre per Definition der kanonischen Ordnung  $v_i$  der am weitesten rechts gelegene Nachbar von  $C_I$  in  $G_{k-1}$ . Allerdings liegt der Elternknoten von  $v_k$  weiter rechts als  $v_i$ .  $\zeta$

Sei nun  $v$  ein Knoten in  $H$ . Es ist zu zeigen:  $v$  ist ein geordneter Knoten, d.h. die Nachbarn von  $v$  bilden im Gegenuhrzeigersinn um  $v$  die vier Blöcke, die in der Definition beschrieben wurden.

Ist  $v = v_1$ , so ist  $v$  geordnet, denn alle seine Nachbarn sind Kinder in  $T$ . Wäre dies nicht der Fall, so wäre  $v$  (als Wurzel von  $T$ ) automatisch mit einem seiner Nichtkind-Nachbarn verwandt. Dies wäre aber ein Widerspruch zu der oben gezeigten Aussage.

Sei nun  $v = v_i$  mit  $i > 1$ . Überlegen wir uns, welche Probleme auftreten könnten:

1. Annahme:  $v$  hat ein Kind  $w = v_k$  mit  $k$  minimal, für das gilt: Es gibt eine Kante  $\{v, x\}$  mit  $x = v_j$  nichtverwandter Nachbar von  $v = v_i$ ,  $j < i$  und  $\{v, w\}$  kommt beim Durchlaufen der zu  $v$  inzidenten Kanten im Gegenuhrzeigersinn um  $v$  noch vor  $\{v, x\}$ .

Dann wird  $w$  von der Kante  $\{v, x\}$  und dem Baumpfad  $(v, \dots, v_1, \dots, x)$  (der über die Wurzel  $v_1$  verläuft) eingeschlossen.

Wir zeigen nun die folgenden Aussagen:

- (i)  $v$  und  $w$  liegen bzgl. der kanonischen Ordnung nicht im selben Intervall.
- (ii)  $w$  hat keine Kinder in  $T$ .
- (iii)  $w = v_k$  hat einen Nachbarn mit kleinerem Index, der nicht auf dem Rand des Graphen  $H_{k-1}$  liegt.

Aus (i) und (iii) ergibt sich dann aber ein Widerspruch zur Definition der kanonischen Ordnung.

(i): Da  $v$  zwei Nachbarn mit niedrigerem Index hat, nämlich seinen Elternknoten und  $x$ , kann  $v$  nur in einem Intervall der Länge 0 liegen.

(ii): Alle Nachfolger von  $w$  müssen im gleichen Intervall  $I_w = [k, k + q]$  wie  $w$  liegen, da sonst ein Nachfolger einen Nachbarn mit niedrigerem Index im Intervall  $I_w$  besitzt. Da der zugehörige Pfad  $C_{I_w}$  nicht auf dem Rand des Graphen  $H_{k+q}$  liegt, wäre das ein Widerspruch zur Definition der kanonischen Ordnung.

Sei  $u$  der letzte Nachfolger im Intervall  $I_w$ . Außer seinem Elternknoten kommen für  $u$  nur Knoten aus  $H_i$  als Nachbarn in Frage (\*), denn ein Knoten mit höherem Index kann kein Nachbar eines Knoten aus  $I_w$  sein. Sonst hätte der Nachbarn aus  $I_w$  einen niedrigeren Index, läge aber nicht auf dem Rand von  $H_{k+q}$ .

Ist  $u \neq w$ , so darf  $u$  per Definition genau einen Nachbarn aus  $H_{k-1}$  haben. Wegen (\*) hat  $u$  außer seinem Elternknoten genau einen Nachbarn  $b \in H_i$ . Damit wird durch das Entfernen der Knoten  $v$  und  $b$  der Pfad  $C_{I_w}$  vom Rest des Graphen getrennt, was ein Widerspruch zur Voraussetzung 'G ist dreifach zusammenhängend' ist.

(iii): Nach (ii) hat  $w$  keine Kinder. Also besteht der Pfad  $C_{I_w}$  nur aus  $w$ . Dann hat  $w$  per Definition mindestens zwei Nachbarn in  $H_{k-1}$ , genauer: in  $H_i$  (wegen (\*)).

Sei  $y \neq x$  einer dieser Nachbarn. Sei  $z_1$  der letzte Knoten im Intervall, das den ersten Knoten  $\tilde{y} \neq y$  auf dem Baumpfad von  $y$  nach  $x$  enthält. Dann hat  $z_1$  per Definition einen nichtverwandten Nachbarn  $\tilde{z}_2$  mit niedrigerem Index als  $\tilde{y}$ . Die zugehörige Kante liegt in  $H_i$ . Sein nun  $K_1$  der Kreis der durch den Pfad  $(z_1, \dots, x, v, \dots, v_1, \dots, \tilde{z}_2)$  (über die Baumkanten und die Kante  $\{v, x\}$ ) und die Kante  $\{z_1, \tilde{z}_2\}$  entsteht. Dieser Pfad liegt nun vollständig in  $H_i$  und je nach Lage der Kante  $\{z_1, \tilde{z}_2\}$  liegen  $v$  und die Wurzel  $v_1$  oder alle Baumknoten, die zwischen  $z_1$  und dem größten gemeinsamen Vorgänger von  $z_1$  und  $\tilde{z}_2$  liegen, im Inneren von  $H_i$ . Der erste Fall kann nicht eintreten, da  $v_1$  ein externer Knoten von  $H$  ist. Im zweiten Fall sind wir fertig, wenn  $y$  einer dieser Zwischenbaumknoten ist. Ist dies nicht der Fall, so finden wir nun aber zum letzten Knoten  $z_2$  des Intervalls, das  $\tilde{z}_2$  enthält, einen Nachbarn  $\tilde{z}_3$  mit niedrigerem Index als  $\tilde{z}_2$ . Schließen wir nun analog, so erhalten wir nach endlich vielen Schritten ein  $\tilde{z}_m$ , das einen niedrigeren Index als  $y$  hat, und somit liegt  $y$  zwischen  $z_1$  und dem größten gemeinsamen Vorgänger von  $z_1$  und  $\tilde{z}_m$ . Das bedeutet aber, im Inneren des Kreises  $K_{m-1} \subset H_i$ , der aus den Pfaden  $(z_1, \dots, x, v, \dots, v_1, \dots, \tilde{z}_m)$  und  $(z_1, \tilde{z}_2, \dots, z_2, \tilde{z}_3, \dots, z_3, \dots, \tilde{z}_m)$  entsteht, liegt  $y$ .

2. Annahme:  $v$  hat einen nichtverwandten Nachbarn  $w$  mit höherem Index für den gilt: Es gibt eine Kante  $\{v, x\}$  mit  $x = v_j$  nichtverwandter Nachbar von  $v = v_i$ ,  $j < i$  und  $\{v, w\}$  kommt beim Durchlaufen der zu  $v$  inzidenten Kanten im Gegenuhrzeigersinn um  $v$  noch vor  $\{v, x\}$ .

Dann wird  $w$  von der Kante  $\{v, x\}$  und dem Baumpfad  $(v, \dots, v_1, \dots, x)$  (der über die Wurzel  $v_1$  verläuft) eingeschlossen. Andererseits existiert aber auch ein Baumpfad  $(w, \dots, v_1)$ . Dieser müßte aber den Kreis (zusätzlich zu dem Schnittpunkt  $v_1$ ) schneiden, da  $x$  einen höheren Index als  $v$  hat. Dies geht aber weder in einer Kante, noch in einem Baumknoten.  $\nabla$

3. Annahme:  $v$  hat ein Kind  $w = v_k$  mit  $k$  minimal, für das gilt: Es gibt eine Kante  $\{v, x\}$  mit  $x = v_j$  nichtverwandter Nachbar von  $v = v_i$ ,  $j > i$  und  $\{v, x\}$  kommt beim Durchlaufen der zu  $v$  inzidenten Kanten im Gegenuhrzeigersinn um  $v$  noch vor  $\{v, w\}$ .

Hier ergibt sich ein Widerspruch analog zur ersten Annahme.

q.e.d.

Nun stellt sich die Frage:

Erhält man aus einem geordneten Spannb Baum eines dreifach zusammenhängenden planar eingebetteten Graphen auch wieder eine kanonische Ordnung?

Die Antwort kann man für triangulierte Graphen geben:

### 3.8 Satz:

Sei  $H$  ein triangulierter Graph und sei  $T$  ein geordneter Spannbaum. Dann ist die Präordnung im Gegenuhrzeigersinn von  $T$  eine kanonische Ordnung von  $H$ .

Beweis:

Sei  $v_1, \dots, v_n$  die Präordnung im Gegenuhrzeigersinn der Knoten von  $T$ . Zunächst folgt direkt aus der Definition des geordneten Baumes und der Präordnung im Gegenuhrzeigersinn, daß die Kante  $\{v_1, v_2\}$  auf dem Rand des Graphen liegt.

Setze nun  $I_j := [j + 2, j + 2]$  für  $j = 1, \dots, n - 2$ .

Dann gilt:

- (i)  $I_1, \dots, I_{n-2}$  sind geordnete Intervalle, die das Intervall  $[3, n]$  partitionieren.
- (ii) Für  $j = 1, \dots, n - 2$  gilt:  $C_j$  hat keine Sehnen in  $H$ .
- (iii) Für  $k = 3, \dots, n$  gilt:  $v_k$  hat mindestens zwei Nachbarn in  $H_{k-1}$ , alle liegen auf dem Rand dieses Teilgraphen.
- (iv) Für  $k = 3, \dots, n$  gilt:  $H_{k-1}$  ist zweifach zusammenhängend.
- (v) Für  $k = 3, \dots, n - 1$  gilt:  $v_k$  hat mindestens einen Nachbarn in  $H \setminus H_k$ .

Haben wir diese fünf Eigenschaften nachgewiesen, ist gezeigt, daß  $v_1, \dots, v_n$  eine kanonische Ordnung von  $H$  bilden.

(i) und (ii) folgen sofort aus der Definition der Intervalle  $I_1, \dots, I_{n-2}$ .

(iii): Wir müssen die folgenden drei Fälle unterscheiden:

1. Fall:  $v_k$  besitzt ein Kind in  $T$ .

Sei  $w$  das Kind von  $v_k$  mit dem niedrigsten Index, also das erste, wenn man die Kinder von  $v_k$  im Gegenuhrzeigersinn um  $v_k$  durchläuft. Nehmen wir nun an,  $v_k$  hätte nur einen Nachbarn mit niedrigerem Index, also seinen Elternknoten  $u$ , so lägen  $u$ ,  $v_k$  und  $w$  zusammen in einer Facette von  $H$ . Damit müßten  $u$  und  $w$  wegen der Triangulierung von  $H$  eine gemeinsame Kante haben. Das würde aber bedeuten,  $u$  und  $w$  wären bzgl.  $T$  verwandt, was im Widerspruch dazu steht, daß  $T$  ein geordneter Spannbaum ist.

2. Fall:  $v_k$  besitzt kein Kind in  $T$  und liegt direkt unter der Wurzel  $v_1$  von  $T$ .

Dann besitzt  $v_1$  mindestens ein weiteres Kind mit niedrigerem Index, nämlich  $v_2$ . Wählt man nun das Kind von  $v_1$  mit dem nächstkleineren Index, so liegt es zusammen mit  $v_k$  in einer Facette. Wegen der Triangulierung von  $H$  haben diese beiden Knoten nun eine Kante in  $H$ . Somit haben wir einen weiteren Nachbarn von  $v_k$  mit niedrigerem Index gefunden.

3. Fall:  $v_k$  besitzt kein Kind in  $T$  und liegt nicht direkt unter der Wurzel  $v_1$  von  $T$ .

Dann besitzt der Elternknoten von  $v_k$  Nachbarn, die einen niedrigeren Index haben als  $v_k$ . Das sind entweder andere Kinder mit niedrigerem Index als  $v_k$  oder eben nichtverwandte Nachbarn mit niedrigerem Index (wie in Fall 1 bzw. 2). Einer dieser Nachbarn liegt nun zusammen mit  $v_k$  in einer Facette, hat also wegen der Triangulierung von  $H$  eine Kante mit  $v_k$ .

(iv): Dies erhält man über folgende Induktion nach  $k$ :

$k = 3$ :

$H_2$  besteht nur aus der Kante  $\{v_1, v_2\}$ , ist also offensichtlich zweifach zusammenhängend.

$k \rightarrow k + 1$ :

Entfernt man  $v_k$ , so ist der Graph  $H_{k-1}$  nach Induktionsvoraussetzung zweifach zusammenhängend, also insbesondere zusammenhängend. Entfernt man einen Knoten aus  $H_k \setminus v_k$ , so ist der Graph  $H_k$  wegen (iii) und der Induktionsvoraussetzung immer noch zusammenhängend.

(v): Wieder müssen wir drei Fälle unterscheiden:

1. Fall:  $v_k$  hat ein Kind in  $T$ . Dann sind wir fertig.
2. Fall:  $v_k$  besitzt kein Kind in  $T$  und liegt direkt unter der Wurzel  $v_1$  von  $T$ .

Da  $v_k$  nicht der letzte Knoten in  $T$  ist, hat  $v_1$  ein weiteres Kind, das einen größeren Index als  $v_k$  hat. Das Kind mit dem nächsthöherem Index liegt mit  $v_k$  in einer Facette, und wegen der Triangulierung von  $H$  haben die beiden eine Kante in  $H$ .

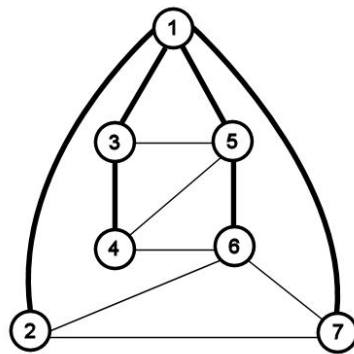
3. Fall:  $v_k$  besitzt kein Kind in  $T$  und liegt nicht direkt unter der Wurzel  $v_1$  von  $T$ .

Dann besitzt der Elternknoten  $u$  von  $v_k$  einen nichtverwandten Nachbarn in  $H$ , der einen höheren Index (als  $v_k$ ) hat. Ansonsten lägen der Elternknoten von  $u$  und  $v_k$  in einer Facette und hätten damit wegen der Triangulierung von  $H$  eine gemeinsame Kante. Das hieße, daß die beiden Knoten verwandt sind. Dies steht aber im Widerspruch dazu, daß  $T$  geordnet ist.

q.e.d.

Setzt man im letzten Satz nur 'dreifach zusammenhängend' voraus, so stimmt diese Aussage nicht mehr:

### 3.9 Beispiel:



3.6

Dieser Graph  $H$  ist dreifach zusammenhängend und die fett gedruckten Kanten bilden einen geordneten Spannbaum  $T$  von  $H$ . Die Präordnung im Gegenuhrzeigersinn von  $T$  ist allerdings keine kanonische Ordnung von  $H$ .

## 4 Ein Algorithmus zur Konstruktion geordneter Paare

Wir haben nun unser Werkzeug definiert. Jetzt stellt sich die Frage:

Kann man, und wenn ja, wie schnell, zu jedem (zusammenhängenden) planaren Graphen ein geordnetes Paar finden?

Die Antwort gibt dieser Abschnitt:

Es wird ein Algorithmus vorgestellt, der in linearer Zeit zu einem gegebenen zusammenhängenden planaren Graphen ein geordnetes Paar berechnet.

Dazu nehmen wir ohne Einschränkung der Allgemeinheit an, daß der gegebene Graph schon mit einer planaren Einbettung versehen ist, die durch eine Datenstruktur repräsentiert wird, in der das Löschen einer Kante und das Verlegen einer Kante in das Innere einer Facette in konstanter Laufzeit vorgenommen werden kann.

Ein Beispiel für eine solche Datenstruktur ist das Repräsentieren der Einbettung durch eine Adjazenzliste, in der jeder Knoten  $v$  eine doppelt verkettete Liste besitzt, die seine Nachbarn in einer Ordnung, die im Gegenuhrzeigersinn um  $v$  verläuft, speichert. Dabei sind je zwei Kopien einer Kante gegenseitig verlinkt. Solch eine Repräsentation erhält man zum Beispiel als ein Nebenprodukt eines Algorithmus von Hopcroft und Tarjan ([4]), der in linearer Zeit testet, ob ein gegebener Graph planar ist.

Wie wir schon in Beispiel 3.2 gesehen haben, existiert nicht zu jeder planaren Einbettung eines zusammenhängenden Graphen ein geordneter Spannbaum. Deswegen wird ein Algorithmus, der ein geordnetes Paar berechnet im Allgemeinen zuerst die planare Einbettung soweit verändern müssen, bis sie einen geordneten Spannbaum zulässt.

Zunächst betrachten wir den folgenden Algorithmus für zweifach zusammenhängende planare Graphen  $G$ . Anhand von Beispielgraphen wird der Ablauf des Algorithmus verdeutlicht.

Ist  $x$  ein Knoten auf dem Rand von  $G$ , so sei  $\text{prev}(G, x)$  bzw.  $\text{next}(G, x)$  der Knoten, der vor bzw. hinter  $x$  liegt, wenn man den Rand von  $G$  im Gegenuhrzeigersinn durchläuft.

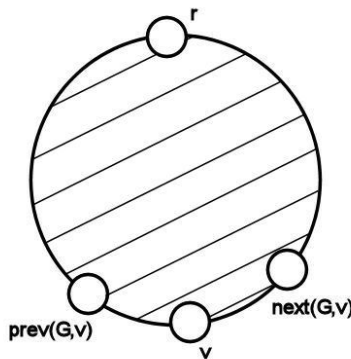
$BLOCK(G, r, v)$  (für zwei beliebige Knoten  $r, v$  auf dem Rand von  $G$ )

– Schritt 1 –

Falls  $G$  nur aus der Kante  $\{r, v\}$  besteht, gibt  $(G, G)$  zurück.

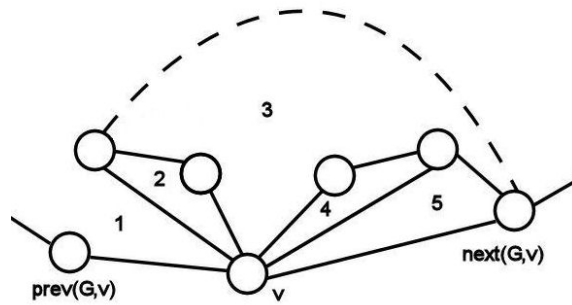


Ansonsten führe aus:



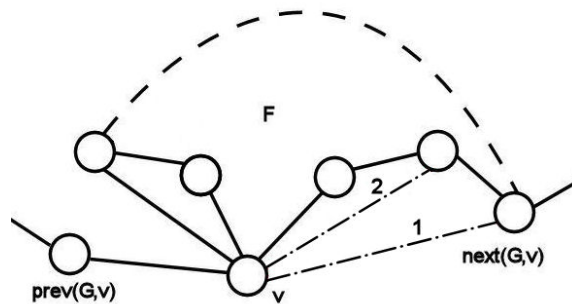
– Schritt 2: –

Durchlaufe die inneren Facetten von  $G$ , die  $v$  enthalten, um  $v$  im Uhrzeigersinn. Beginne bei der Facette, die die Kante  $\{v, \text{prev}(G, v)\}$  enthält.



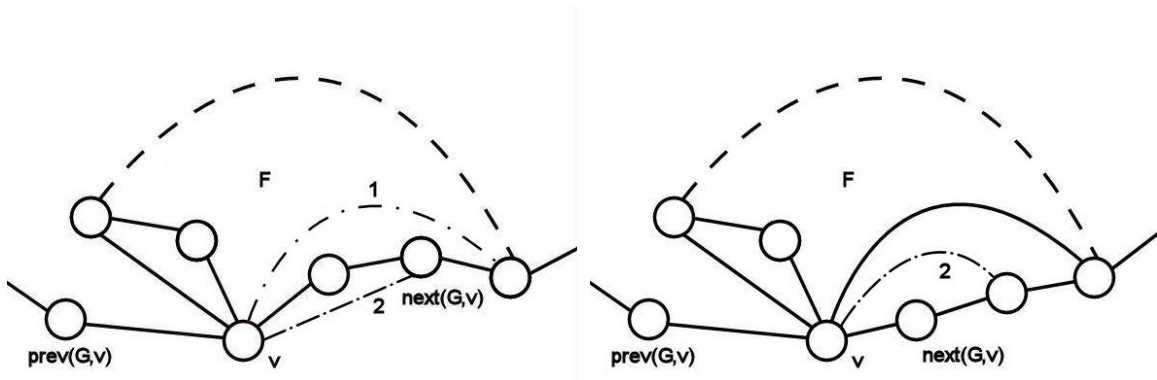
Für jede dieser Facetten  $F$  führe aus:

Betrachte die im Gegenuhrzeigersinn um  $v$  vor  $F$  liegenden Kanten  $\{v, x\}$  mit  $x \in F$ .



Verändere die planare Einbettung von  $G$  folgendermaßen:

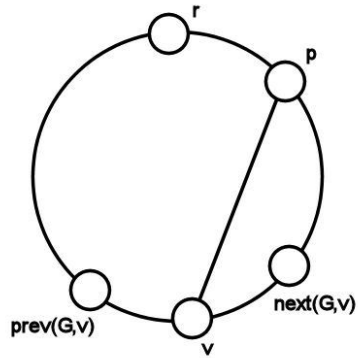
Verlege alle diese Kanten in das Innere von  $F$ .



– Schritt 3: –

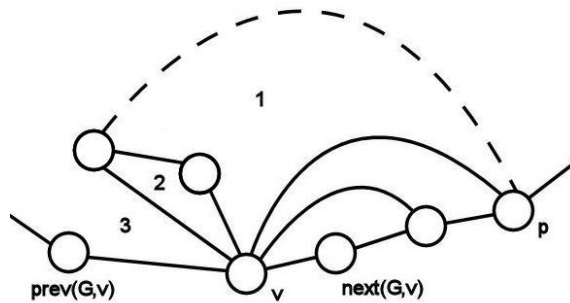
Sei  $p$  der Knoten in  $G$  mit folgenden Eigenschaften:

- $p$  ist ein Nachbar von  $v$ ;
- $p$  liegt auf dem Pfad  $(r, v)$ , wenn man den Rand von  $G$  im Uhrzeigersinn durchläuft, am nächsten bei  $r$ .



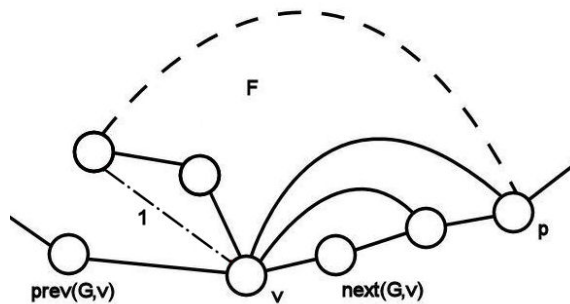
- Schritt 4: -

Durchlaufe die inneren Facetten von  $G$ , die  $v$  enthalten und hinter  $\{v, p\}$  liegen, um  $v$  im Gegenuhrzeigersinn. Beginne bei der Facette, die die Kante  $\{v, p\}$  enthält.



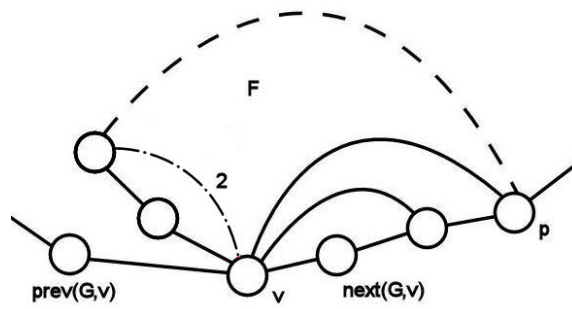
Für jede dieser Facetten  $F$  führe aus:

Betrachte die im Gegenuhrzeigersinn um  $v$  nach  $F$  liegenden Kanten  $\{v, x\}$  mit  $x \in F$ .

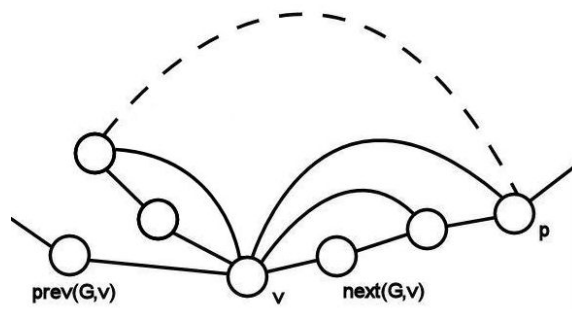


Verändere die planare Einbettung von  $G$  folgendermaßen:

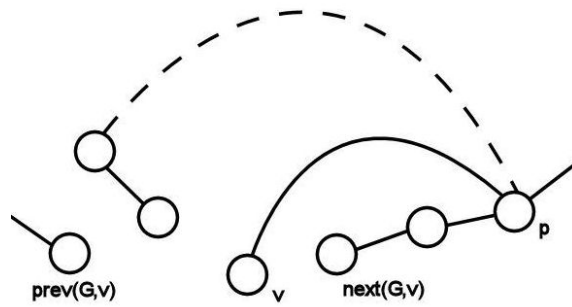
Verlege alle diese Kanten in das Innere von  $F$ .



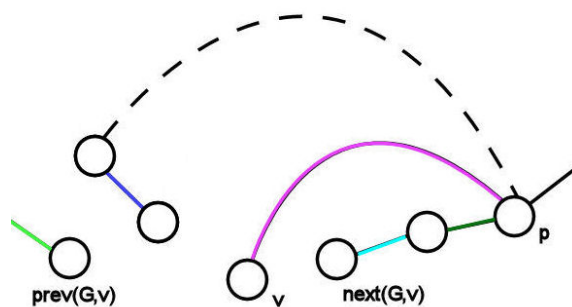
- Schritt 5: -



Sei  $G'$  der Graph, der durch Entfernen aller zu  $v$  inzidenten Kanten außer  $\{v, p\}$  entsteht.



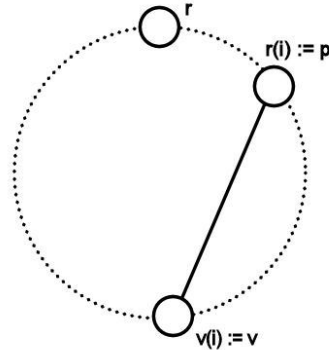
Bestimme die Zweifach-Zusammenhangskomponenten  $G_i$  ( $1 \leq i \leq K$ ) von  $G'$ .



Für  $i = 1$  bis  $K$  führe aus:

Sei  $r_i$  der Knoten von  $G_i$ , der  $r$  am nächsten liegt.

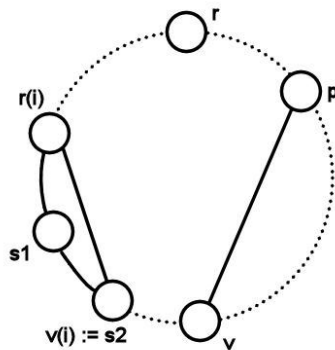
1. Fall.  $G(i) = \{v, p\}$ : setze  $v_i := v$ .



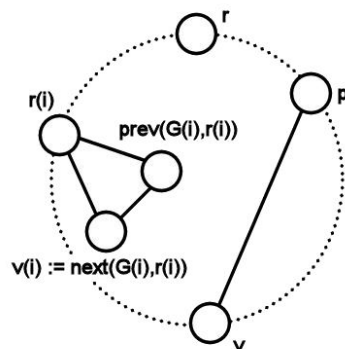
2. Fall.  $G_i$  und  $\text{prev}(G, v)$  liegen auf der selben Seite von  $\{v, p\}$  in  $G$ :

Sei  $S$  die Menge aller Knoten  $\neq r_i$ , die sowohl auf dem Pfad  $(r, v)$ , wenn man den Rand von  $G$  im Gegenuhrzeigersinn durchläuft, als auch auf dem Rand von  $G_i$  liegen.

Falls  $S \neq \emptyset$ , sei  $v_i$  der *letzte* Knoten in  $S$ , wenn man die Knoten im Gegenuhrzeigersinn auf dem Rand von  $G_i$  durchläuft.



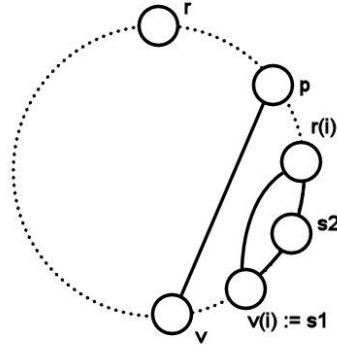
Ansonsten: Setze  $v_i := \text{next}(G_i, r_i)$ .



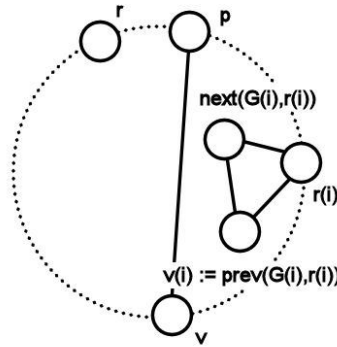
3. Fall.  $G_i$  und  $\text{next}(G, v)$  liegen auf der selben Seite von  $\{v, p\}$  in  $G$ :

Sei  $S$  die Menge aller Knoten  $\neq r_i$ , die sowohl auf dem Pfad  $(r, v)$ , wenn man den Rand von  $G$  im Uhrzeigersinn durchläuft, als auch auf dem Rand von  $G_i$  liegen.

Falls  $S \neq \emptyset$ , sei  $v_i$  der *erste* Knoten in  $S$ , wenn man die Knoten im Gegenuhrzeigersinn auf dem Rand von  $G_i$  durchläuft.



Ansonsten: Setze  $v_i := \text{prev}(G_i, r_i)$ .



$BLOCK(G_i, r_i, v_i)$

– Schritt 7: –

Gib  $(H, T)$  zurück, wobei  $T$  die Vereinigung der  $T_i$  sei, und  $H$  aus  $G$  folgendermaßen entstehe:

Setze  $H := G$ .

Für  $i = 1$  bis  $K$  führe aus:

Ersetze  $G_i$  durch  $H_i$ .

Wenden wir uns zuerst der Korrektheit des Algorithmus zu:

#### 4.1 Lemma:

Sei  $G$  ein zweifach zusammenhängender planar eingebetteter Graph, und seien  $r$  und  $v$  zwei beliebige verschiedene Knoten auf dem Rand von  $G$ . Dann gibt  $BLOCK(G, r, v)$  ein geordnetes Paar (bzgl.  $r$ ) von  $G$  zurück.

Beweis:

Sei  $(H, T)$  das von  $BLOCK(G, r, v)$  zurückgegebene Paar. Wir zeigen die folgenden Aussagen in Abhängigkeit von  $G, H, T, r$  und  $v$ , und per Induktion über die Anzahl der Kanten in  $G$ :

1. Jeder externe Knoten von  $G$  bleibt extern in  $H$ , und die Reihenfolge auf dem Rand bleibt bestehen.
2. Für jeden Nachbarn  $x \neq p$  von  $v$  in  $H$  gilt: Falls  $x$  und  $\text{prev}(H, v)$  (bzw.  $\text{next}(H, v)$ ) auf der selben Seite von  $\{v, p\}$  in  $H$  liegen, dann liegt  $\{v, x\}$  auf der ersten (bzw. letzten) inneren Facette von  $H$ , die  $v$  und  $x$  enthält, wenn man die inneren Facetten von  $H$ , angefangen bei derjenigen, die die Kante  $\{v, \text{next}(H, v)\}$  enthält, im Gegenurzeigersinn durchläuft.

3.  $T$  ist ein Spannbaum von  $H$  mit Wurzel  $r$  und der Eigenschaft, daß für alle Knoten  $u$  auf dem Pfad  $(r, v)$ , der entsteht wenn man von  $r$  nach  $v$  im Gegenuhrzeigersinn (bzw. Uhrzeigersinn) auf dem Rand läuft, genau eine der folgenden Bedingungen erfüllt ist: (i)  $u$  ist ein Blatt von  $T$ , oder (ii)  $\text{next}(H, u)$  (bzw.  $\text{prev}(H, u)$ ) ist das Kind von  $u$  in  $T$  mit dem niedrigsten (bzw. höchsten) Index.
4.  $H$  ist eine planare Einbettung von  $G$ .
5.  $T$  ist ein geordneter Spannbaum von  $H$  mit Wurzel  $r$ .

Haben wir die Aussagen 4 und 5 gezeigt, so sind wir fertig.

Nun führen wir eine Induktion über die Anzahl der Kanten von  $G$ :

*Induktionsanfang:* Besteht  $G$  nur aus einer einzigen Kante  $\{r, v\}$ , so gilt nach Schritt 1 des Algorithmus *BLOCK*:  $H = T = G$ . Offensichtlich gelten die Aussagen 1 – 5. Damit ist für jede dieser Aussagen der Induktionsanfang erfüllt.

Seien  $G_1 \dots, G_k$  die Zweifach-Zusammenhangskomponenten von  $G'$ . Wir bezeichnen nun mit der Aussage  $j(i)$  die Aussage  $j \in \{1, 2, 3, 4, 5\}$  bezogen auf  $G_i, H_i, T_i, r_i$  und  $v_i$  für  $i \in \{1, \dots, k\}$ . Nach Induktionsvoraussetzung sind alle diese Aussagen erfüllt.

Wir zeigen nun den *Induktionsschritt* für die Aussagen 1 – 5 bezogen auf  $G, H, T, r$  und  $v$ :

1. Während des Ausführens von  $\text{BLOCK}(G, r, v)$  wird, wenn man die darin vorkommenden nachfolgenden Aufrufe von  $\text{BLOCK}$  außer acht läßt, die Einbettung von  $G$  nur durch das Umlegen von Kanten in das Innere von inneren Facetten von  $G$  während der Schritte 2 und 4 verändert. Dabei bleiben externe Knoten extern und die Reihenfolge auf dem Rand bleibt erhalten.  $H$  entsteht nun durch das Ersetzen der Zweifach-Zusammenhangskomponenten  $G_i$  von  $G'$  mit den zugehörigen  $H_i$ . Nach Induktionsannahme ist die Aussage 1 für alle diese  $H_i$  richtig, also auch für  $H$ .
2. Bezeichnen wir zuerst einmal mit Aussage 2' die Aussage, die aus Aussage 2 entsteht, wenn man jedes  $H$  durch ein  $G$  ersetzt. Betrachtet man die Schritte 2 und 4 des Algorithmus, so sieht man, daß nach Schritt 4 für  $G$  die Aussage 2' wahr ist. Da  $H$  aus  $G$  durch das Ersetzen der  $G_i$  durch die  $H_i$  entsteht, erhält man mit Hilfe der Aussagen 1(i), daß die relative Ordnung der zu  $v$  inzidenten Kanten und der Facetten, die  $v$  enthalten, in  $G$  und  $H$  gleich bleiben. Damit folgt die Aussage 2 aus der Aussage 2'.
3. Für alle  $i \in \{1, \dots, k\}$  impliziert die Aussage 3(i), daß  $T_i$  ein Spannbaum von  $H_i$  mit Wurzel  $r_i$  ist. Da  $H_1, \dots, H_k$  kantendisjunkt sind, und jeder Knoten von  $H$  zu einem  $H_i$  gehört, wissen wir, daß  $T$  — die Vereinigung der  $T_i$  — ein Spannbaum von  $H$  mit Wurzel  $r$  ist.  
Da  $v$  ein Blatt von  $T$  ist, ist die geforderte Eigenschaft für  $v$  erfüllt. Sei nun  $x$  ein von  $v$  verschiedener externer Knoten von  $H$ . Falls  $\{x, v\}$  keine externe Kante von  $H$  ist, die zu  $H \setminus T$  gehört, so wird die Eigenschaft für  $x$  von den Aussagen 3(i) für jedes  $i$  mit  $x \in H_i$  garantiert. Ansonsten ist  $x$  entweder  $\text{prev}(H, v)$  oder  $\text{next}(H, v)$ . Sei  $H_l$  die Zweifach-Zusammenhangskomponente von  $H'$ , die  $x$  enthält. Dann gilt:  $x = v_l$  (vgl. Schritt 6). Nach Aussage 3(l) ist  $x$  ein Blatt von  $T_l$ .
4. Bei den Schritten 2 und 4 wird eine Kante  $\{v, x\}$  nur dann in das Innere einer Facette verlegt, wenn diese Facette die beiden Knoten enthält. Somit ist die Einbettung von  $G$  nach Schritt 4 immernoch planar. Aufgrund der Aussagen 4(i) für alle  $i \in \{1, \dots, k\}$  sind alle  $H_i$  planare Einbettungen der Zweifach-Zusammenhangskomponenten  $G_i$  von  $G'$ . Beim Ersetzen der  $G_i$  durch die  $H_i$  bleibt die Planarität aufgrund der Aussagen 1(i) für alle  $i \in \{1, \dots, k\}$  erhalten.
5. Jeder Nachbar von  $r$  in  $H$  ist ein Kind von  $r$  in  $T$ , denn es gilt  $r = r_i$  für alle  $i$  mit  $r \in H_i$ . Nach Induktionsannahme sind die Aussagen 5(i) für alle  $i$  erfüllt, somit ist jeder Nachbar von  $r$  ein Kind von  $r$  in einem der  $T_i$ , also auch in  $T$ . Damit ist  $r$  bzgl.  $T$  geordnet.  
Sei nun  $x$  ein von  $r$  verschiedener Knoten. Sei  $H'$  der Graph, den man aus  $H$  durch das Entfernen aller zu  $v$  in  $H \setminus T$  inzidenten Kanten erhält.  $H'$  ist offensichtlich eine planare Einbettung von  $G'$ , und alle  $H_i$  sind Zweifach-Zusammenhangskomponenten von  $H'$ . Sei  $I_x$  die Menge aller Indizes  $i$  mit  $x \in H_i$ . Wegen  $x \neq r$  gibt es einen Index  $l \in I_x$  mit  $x \neq r_l$  und  $x = r_i$  für alle  $i \in I_x \setminus \{l\}$ :

Sei  $l \in I_x$  der Index  $i$ , für den  $r_i$  am nächsten bei  $r$  liegt.

Ist  $x \in H_l$  intern, so ist  $x \neq r_l$ , und damit  $I_x = \{l\}$ .

Ist  $x \in H_l$  extern, so ist ebenfalls  $x \neq r_l$ , denn sonst gäbe es wegen  $r \neq x = r_l$  ein  $l_0$  mit  $x \in H_{l_0}$ , und  $r_{l_0}$  läge näher bei  $r$  als  $r_l$ , da es auf dem Baumpfad von  $r_l$  zur Wurzel  $r$  läge. Ist nun  $x \notin H_i$  für alle  $i \neq l$ , so sind wir fertig. Ansonsten ist  $x = r_i$  für alle  $i \neq l$  mit  $x \in H_i$ , denn wäre  $x \neq r_{l_1}$  für ein solches  $l_1 \in I_x \setminus \{l\}$ , so könnte  $r_{l_1}$  nicht in  $H_l$  liegen, sonst wäre wegen  $x \neq r_{l_1} \in H_l \cap H_{l_1}$  schon  $H_l = H_{l_1}$ . Allerdings wären sowohl  $r_l$  als auch  $r_{l_1}$  Vorgänger von  $x$  in  $T$ , d.h. entweder würde  $l > l_1$  oder  $l_1 > l$  gelten. Der erste Fall wäre ein Widerspruch zur Wahl von  $l$ , der zweite Fall dazu, daß  $r_{l_1}$  nicht in  $H_l$  liegen könnte.

Wir zeigen zuerst, daß  $x$  ein bzgl.  $T$  geordneter Knoten in  $H'$  ist. Falls  $\#I_x = 1$  ist, folgt dies aus Aussage 5( $l$ ), da dann  $x$  keine Nachbarn in anderen Zweifach-Zusammenhangskomponenten von  $H'$  hat. Falls  $\#I_x \geq 2$ , folgt aus den Aussagen 5( $i$ ) für alle  $i \in I_x \setminus \{l\}$ , wegen  $x = r_i$  für diese Indizes  $i$ , daß alle Nachbarn von  $x$  in  $\bigcup_{i \in I_x \setminus \{l\}} H_i$  Kinder von  $x$  in  $T$  sind. Aufgrund der Aussage 3( $l$ ), und da  $x$  in  $H_l$  bzgl.  $T_l$  geordnet ist, wissen wir, daß alle Kinder von  $x$  in  $T$  aufeinander folgen, wenn man die Nachbarn von  $x$  — genauer: die zu  $x$  inzidenten Kanten — im Gegenuhrzeigersinn um  $x$  betrachtet. Somit folgt ebenfalls, weil  $x$  in  $H_l$  bzgl.  $T_l$  geordnet ist, daß  $x$  in  $H'$  bzgl.  $T$  geordnet ist.

Nun zeigen wir, daß, falls  $\{v, x\}$  eine Kante von  $H \setminus H'$  ist, also  $x \neq p$  ist,  $v$  und  $x$  nicht verwandt bzgl.  $T$  sind. Falls der Index von  $x$  größer als der von  $v$  ist, so folgt dies schon aus der Tatsache, daß  $v$  ein Blatt von  $T$  ist. Ist der Index von  $x$  niedriger als der von  $v$ , so zeigen wir diese Aussage indirekt. Wir nehmen also an, daß  $x$  ein Vorgänger von  $v$  in  $T$  ist. Wegen  $\{v, x\} \in H \setminus T$ , und da  $p$  der Elternknoten von  $v$  ist, folgt, daß  $x$  auch ein Vorgänger von  $p$  in  $T$  ist. Sei  $P$  der Baumpfad zwischen  $r$  und  $p$ . Dann liegt  $x$  in  $P$ . Sei  $y$  der Knoten von  $H_l$ , der am nächsten bei  $p$  in  $P$  liegt. Es gilt:  $y$  ist ein externer Knoten von  $H$ , also auch von  $H_l$ , und liegt auch dem Pfad von  $r$  nach  $p$ , der entsteht, wenn man den Rand von  $H$  im Uhrzeigersinn durchläuft, sowie auf dem Pfad von  $r_l$  nach  $x$ , der entsteht, wenn man den Rand von  $H_l$  im Uhrzeigersinn durchläuft (\*):

Nehmen wir an,  $y$  ist interner Knoten von  $H$ . Dann gibt es Knoten  $w_1$  und  $w_2$ , die auf dem Uhrzeigersinn-Randpfad von  $r$  nach  $p$  liegen, mit der zusätzlichen Eigenschaft, daß  $w_1$  vor und  $w_2$  nach  $y$  in  $P$  liegt, und daß  $\text{prev}(H, w_1)$  und  $\text{next}(H, w_2)$  nicht in  $P$  liegen. Somit liegt aber der Kreis, der sich aus dem Randabschnitt  $(w_1, w_2)$  und dem Teilpfad von  $P$  zwischen diesen beiden Knoten zusammensetzt, ganz in  $H'$ . Da  $x$  und  $y$  auf diesem Kreis liegen, liegt der Kreis schon komplett in der Zweifach-Zusammenhangskomponente  $H_l$  von  $H'$ . Aber  $w_2$  liegt in  $P$  näher bei  $p$  als  $y$ . Dies ist ein Widerspruch zur Wahl von  $y$ .

$x$  ist ein externer Knoten von  $H_l$ , da  $v \notin H_l$ , aber die Kante  $\{v, x\}$  existiert.

Man kann nun leicht sehen, daß die Annahme (\*) gelte nicht, dazu führen würde, daß  $x$  auf dem Uhrzeigersinn-Randpfad von  $r$  nach  $p$  liegen müsste. Dies wäre aber wegen  $\{v, x\} \in H$  ein Widerspruch zur Wahl von  $p$ . (\*\*)

Wegen (\*\*) müssen aber auch  $x$  und  $y$  verschieden sein. Also ist  $x$  kein Blatt in  $T_l$ . Sei nun  $z := \text{prev}(H_l, x)$ . Nach Aussage 3( $l$ ) wissen wir, daß  $z$  ein Kind von  $x$  in  $T_l$  ist. Wegen  $x \neq r_l$  und (\*) liegen  $z$  und  $y$  auf verschiedenen Seiten des Baumpfades von  $r_l$  nach  $x$  in  $T_l$ . Dies steht bei genauerer Betrachtung im Widerspruch dazu, daß  $z$  nach 3( $l$ ) der Nachbar von  $x$  mit dem höchsten Index ist.

Da  $v$  ein Blatt von  $T$  ist, ist  $v$  in  $H$  bzgl.  $T$  geordnet. Wir müssen nur noch zeigen, daß auch alle von  $p$  verschiedenen Nachbarn  $x$  von  $v$  in  $H$  bzgl.  $T$  geordnet sind. Sei  $z_1$  (bzw.  $z_2$ ) der Nachbar von  $x$ , der vor (bzw. nach)  $v$  in der Ordnung der Nachbarn um  $x$  liegt, wenn man im Gegenuhrzeigersinn um  $x$  herumläuft. Es genügt zu zeigen, daß, falls der Index von  $x$  niedriger (bzw. höher) als der von  $v$  ist,  $z_2$  (bzw.  $z_1$ ) in  $B_1(x)$  oder  $B_4(x)$  (bzw.  $B_1(x)$  oder  $B_2(x)$ ) von  $H'$  bzgl.  $T$  liegt.

Falls der Index von  $x$  niedriger als der von  $v$  ist, dann folgt aus Aussage 2, daß  $z_2 = \text{next}(H_l, x)$  ist. Denn: Andernfalls würde  $z_2$  nicht zu  $H_l$  gehören. Damit müsste es aber in  $H$  einen weiteren Pfad von  $z_2$  nach  $\text{next}(H_l, x)$  geben. Dieser müsste über  $v$  verlaufen, und das würde bedeuten, daß  $x$  und  $v$  in einer inneren Facette liegen, die nicht die Kante  $\{v, x\}$  enthält, und die im Gegenuhrzeigersinn um  $v$  noch vor der ersten Facette liegt, die diese Kante enthält. Dies steht im

Widerspruch zur Aussage 2.

Nach Schritt 6 des Algorithmus liegt  $x$ , wegen  $\{v, x\} \in H$ , auf dem Pfad von  $r_l$  nach  $v_l$ , wenn man am Rand von  $H_l$  im Uhrzeigersinn entlang läuft. Dies, zusammen mit Aussage 3( $l$ ) angewendet auf  $z_2$ , liefert uns  $z_2 \in B_1(x)$  oder  $z_2 \in B_4(x)$  in  $H'$  bzgl.  $T$ .

Analog bekommen wir, falls der Index von  $x$  höher ist als der von  $v$ , daß  $z_1 = \text{prev}(H_l, x)$  ist und  $x$  auf dem Pfad von  $r_l$  nach  $v_l$  liegt, wenn man am Rand von  $H_l$  im Gegenuhrzeigersinn entlang läuft, und damit,  $z_1 \in B_1(x)$  oder  $z_1 \in B_2(x)$  in  $H'$  bzgl.  $T$ .

q.e.d.

Nun werden wir die Laufzeit des Algorithmus betrachten:

#### 4.2 Lemma:

Sei  $G$  ein zweifach zusammenhängender planar eingebetteter Graph mit  $n$  Knoten, und seien  $r$  und  $v$  zwei beliebige, verschiedene Knoten auf dem Rand von  $G$ . Dann liegt die Laufzeit von  $BLOCK(G, r, v)$  in  $\mathcal{O}(n)$ .

Beweis:

Beim Ausführen von  $BLOCK(G, r, v)$  wird eine Reihe von weiteren  $BLOCK$ -Aufrufen ausgeführt. Ein Knoten  $w$  kann dabei höchstens zweimal als Parameter 'v' auftreten. Einmal für einen Teilgraphen der nicht die Form  $\{r_w, w\}$  hat (Aufruf vom Typ I), und einmal für einen Teilgraphen dieser Form (Aufruf vom Typ II). Denn: Ein Aufruf vom Typ I mit  $w$  als Parameter 'v' impliziert einen späteren Aufruf vom Typ II (mit  $w$  als Parameter 'v'). Außerdem wird kein weiterer Aufruf vom Typ I (mit  $w$  als Parameter 'v') durchgeführt, und nach einem Aufruf vom Typ II wird kein Teilgraph mehr betrachtet, der  $w$  enthält.

Bei einem Aufruf vom Typ II ist die Laufzeit  $\mathcal{O}(1)$ .

Sei nun  $l$  die Anzahl aller Aufrufe vom Typ I während des Ausführens von  $BLOCK(G, r, v)$ . Sei für  $j \in \{1, \dots, l\}$   $BLOCK(G_j, r_j, v_j)$  der  $j$ -te Aufruf vom Typ I, also mit  $G_j \neq \{r_j, v_j\}$ . Dann ist  $G_1 = G$ ,  $r_1 = r$  und  $v_1 = v$ , und es gilt:  $v_j \neq v_{j'}$  für  $j \neq j'$ . Damit folgt auch  $l \leq n$ .

Sei  $B_j$  die Menge aller Kanten in  $G$ , die auf dem Rand einer inneren Facette von  $G_j$  liegen, die  $v_j$  enthält. Sei  $t_j$  die Laufzeit von  $BLOCK(G_j, r_j, v_j)$  ohne die nachfolgenden Aufrufe von  $BLOCK$ . Betrachtet man den Algorithmus, so stellt man fest, daß folgendes gilt:  $t_j = \mathcal{O}(|B_j|)$ .  $BLOCK$  kann nun so implementiert werden, daß die Laufzeit  $\sum_{j=1}^l t_j \in \mathcal{O}(\sum_{j=1}^l |B_j|)$  ist.

$G$  hat  $\mathcal{O}(n)$  Kanten, also genügt es zu zeigen, daß jede Kante  $\{x, y\}$  in maximal zwei der Mengen  $B_j$  liegt: Sei  $j_1$  der kleinste Index  $j$  mit  $\{x, y\} \in B_j$ .

1. Fall:  $v_{j_1} \in \{x, y\}$ .

Dann ist  $j_1$  der einzige derartige Index, da die Kante dann noch höchstens einmal betrachtet wird: Sie wird 'entfernt', falls  $p_{j_1} \notin \{x, y\}$ , andernfalls wird später nur noch ein Aufruf vom Typ II durchgeführt, der ja hier nicht gezählt wird.

2. Fall:  $v_{j_1} \notin \{x, y\}$ .

Jede Kante liegt auf dem Rand von genau zwei Facetten von  $G$ . Wir wissen, daß  $\{x, y\}$  auf dem Rand einer inneren Facette von  $G_{j_1}$  liegt. Beim Entfernen aller zu  $v_{j_1}$  inzidenten Kanten, die nicht  $\{v_{j_1}, p_{j_1}\}$  sind, werden alle inneren Facetten, die  $v_{j_1}$  enthalten, mit der äußeren Facette von  $G_{j_1}$  vereinigt. Somit sind alle Kanten aus  $B_{j_1}$  nun externe Kanten.

Enthält auch die andere Facette von  $\{x, y\}$  den Knoten  $v_{j_1}$  (\*), so besteht die Zweifach-Zusammenhangskomponente von  $G'_{j_1}$ , die  $\{x, y\}$  enthält nur aus  $\{x, y\}$ . Damit liegt  $\{x, y\}$  in keinem anderen  $B_j$ .

Ansonsten liegt die externe Kante  $\{x, y\}$  auf dem Rand einer inneren Facette einer größeren Zweifach-Zusammenhangskomponente von  $G'_{j_1}$ . Sei dann  $j_2$  der kleinste Index  $j > j_1$  mit  $\{x, y\} \in B_j$ . Nun können wir wieder unsere Fallunterscheidung machen, wobei im 2. Fall (\*) gilt, da ja nun  $\{x, y\}$  schon extern ist.

q.e.d.

Mit Hilfe des Algorithmus *BLOCK* können wir auch geordnete Paare von beliebigen zusammenhängenden planaren Graphen  $G$  berechnen:

$PAIR(G)$

Seien  $G_1, \dots, G_K$  die Zweifach-Zusammenhangskomponenten von  $G$ .

Sei  $r$  ein externer Knoten von  $G_1$ .

Für alle  $i \in \{1, \dots, K\}$  sei  $r_i$  der Knoten von  $G_i$  mit dem geringsten Abstand zu  $r$  in  $G$

Für  $i = 1$  bis  $K$ , führe aus:

Setze  $(H_i, T_i) := BLOCK(G_i, r_i, next(G_i, r_i))$ .

Gib  $(H, T)$  zurück, wobei  $H$  die Vereinigung der  $H_i$  und  $T$  die Vereinigung der  $T_i$  sei.

Dieser Algorithmus verhilft uns nun also zu folgendem

#### 4.3 Satz:

Zu einem gegebenen zusammenhängenden planaren Graphen mit  $n$  Knoten kann man in  $\mathcal{O}(n)$  ein geordnetes Paar berechnen.

Beweis:

Zur Korrektheit:

Es gilt:  $r_1 = r$  und für  $i = 2, \dots, K$  ist  $r_i$  ein externer Knoten von  $G_i$  und  $G \setminus \{r_i\}$  ist unzusammenhängend. Denn sei  $i \in \{2, \dots, K\}$ , dann liegt  $r_i$  auf dem Rand von  $G_i$ , und von ihm aus geht ein Pfad nach  $r$  in  $G$ , der über keinen weiteren Knoten aus  $G_i$  führt. Angenommen,  $G \setminus \{r_i\}$  ist zusammenhängend. Dann gibt es von allen anderen Knoten aus  $G_i$  aus einen weiteren Pfad zu  $r$  in  $G$ . Damit würde  $r$  zur Zweifach-Zusammenhangskomponente  $G_i$  gehören.  $\zeta$  zu  $i \neq 1$ .

Nach Lemma 4.1 ist  $H_i$  eine planare Einbettung von  $G_i$  für alle  $i \in \{1, \dots, K\}$ , somit ist  $H = \bigcup_{i=1}^K H_i$  eine planare Einbettung von  $G$ . Wegen dem selben Lemma ist  $T_i$  geordneter Spannbaum von  $H_i$  mit Wurzel  $r_i$  für  $i = 1, \dots, K$ , und damit  $T = \bigcup_{i=1}^K T_i$  geordneter Spannbaum von  $H$  mit Wurzel  $r = r_1$ .

Zur Laufzeit:

Sei  $n_i$  die Anzahl der Knoten in  $H_i$ . Es gilt:  $\sum_{i=1}^K n_i \in \mathcal{O}(n)$ .

Nach Lemma 4.2 gilt:  $BLOCK(G_i, r_i, next(G_i, r_i))$  hat Laufzeit  $\mathcal{O}(n_i)$ . Damit hat  $PAIR(G)$  eine Laufzeit in  $\mathcal{O}(\sum_{i=1}^K n_i) = \mathcal{O}(n)$ .

q.e.d.

## 5 Eine Anwendung beim Zeichnen von Graphen

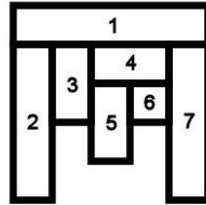
In diesem Abschnitt werden wir auf eine Anwendung der geordneten Bäume beim Zeichnen von Graphen eingehen. Wir zeigen, wie man in  $\mathcal{O}(n)$  eine  $(n-1) \times \lfloor \frac{2n+1}{3} \rfloor$ -2-Sichtbarkeitsdarstellung einen triangulierten Graphen  $G$  mit  $n$  Knoten erhält.

Dabei ist eine 2-Sichtbarkeitsdarstellung eines Graphen eine Zeichnung des Graphen, in der die Knoten

Rechtecke  $\geq 1 \times 1$  mit ganzzahligen Eckkoordinaten sind, und zwei Knoten genau dann eine gemeinsame Kante haben, wenn sie sich die zugehörigen Rechtecke horizontal oder vertikal 'sehen', d.h. es gibt in einer der beiden Richtungen eine Verbindungsstrecke zwischen den Rechtecken, die kein weiteres Rechteck schneidet.

### 5.1 Beispiel:

Die Abbildung 5.1 zeigt ein 2-Sichtbarkeitsdarstellung eines triangulierten Graphen.



5.1

Fößmeier, Kant und Kaufmann entwickelten einen Algorithmus ([3]), der eine  $x \times y$ -2-Sichtbarkeitsdarstellung eines planar eingebetteten Graphen mit  $n$  Knoten, für die  $x + y \leq 2n$  gilt, in  $\mathcal{O}(n)$  berechnet. Außerdem konstruierten sie einen triangulierten Graphen mit  $n$  Knoten, bei dem eine  $x \times y$ -2-Sichtbarkeitsdarstellung folgende Voraussetzungen erfüllen muß:

$$x + y \geq (n - 1) + \lfloor \frac{2n + 1}{3} \rfloor \text{ und } \min\{x, y\} \geq \lfloor \frac{2n + 1}{3} \rfloor.$$

Somit ist der hier vorgestellte Algorithmus, was die hier angegebenen unteren Grenzen angeht, worst-case-optimal.

Betrachten wir nun also diesen Algorithmus:

Sei  $G$  ein triangulierter planar eingebetteter Graph mit  $n$  Knoten. Sei  $T$  ein geordneter Spannbaum von  $G$  und sei  $v_1, \dots, v_n$  die Präordnung im Gegenuhrzeigersinn der Knoten von  $T$ .

*DRAW*( $G, T$ )

Zeichne  $v_1$  als ein Quadrat mit Seitenlänge 1.

Für  $i = 2, \dots, n$  führe aus:

Falls  $v_i$  nicht das erste Kind seines Elternknotens in  $T$  ist, verlängere jeden Vorgänger von  $v_i$  in  $T$  um eine Einheit nach rechts.

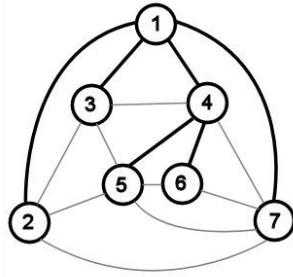
Zeichne  $v_i$  als Quadrat der Seitenlänge 1 (rechts neben den anderen Kindern) unter seinen Elternknoten in  $T$ .

Verlängere  $v_i$  und jeden Nachbarn  $v_j$  in  $G$  mit  $j < i$ , der nicht sein Elternknoten in  $T$  ist, soweit nach unten, bis sie sich horizontal sehen.

Das nächste Beispiel soll die Vorgehensweise des Algorithmus veranschaulichen:

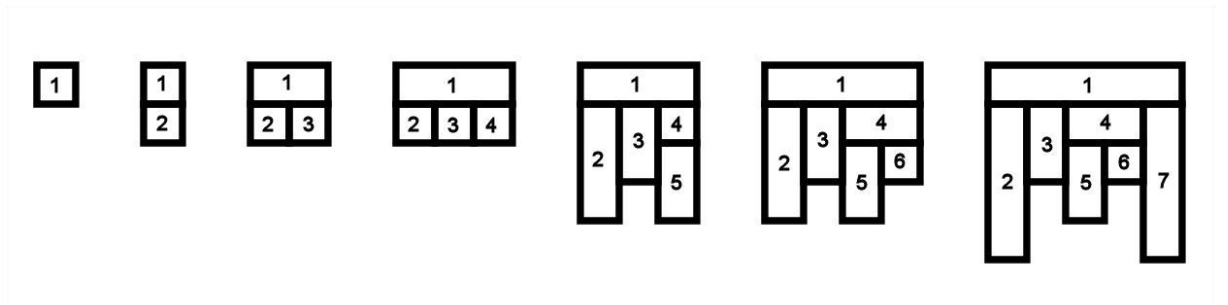
### 5.2 Beispiel:

In der Abbildung 5.2 sehen wir den triangulierten Graphen  $G$  und den zugehörigen geordneten Spannbaum  $T$ , auf die der Algorithmus angewendet werden soll.



5.2

Die Abbildung 5.3 zeigt uns nun, wie die einzelnen Schritte bei Ausführen des Algorithmus  $DRAW(G, T)$  aussehen.



5.3

Die folgenden Ergebnisse sollen hier nicht bewiesen werden. Die Beweise können alle in [1] nachgelesen werden.

**5.3 Lemma:**

Der Algorithmus  $DRAW(G, T)$  erzeugt eine  $x \times y$ -2-Sichtbarkeitsdarstellung von  $G$  mit  $x \leq n - 1$  und  $y$  ist gleich der Anzahl der Blätter von  $T$ .

**5.4 Lemma:**

Man kann in Linearzeit einen geordneten Spannbaum von  $G$  mit  $\lfloor \frac{2n+1}{3} \rfloor$  oder weniger Blättern berechnen.

**5.5 Satz:**

Eine  $(n - 1) \times \lfloor \frac{2n+1}{3} \rfloor$  2-Sichtbarkeitsdarstellung eines triangulierten Graphen mit  $n$  Knoten kann in  $\mathcal{O}(n)$  berechnet werden.

Weitere Anwendungen der geordneten Spannbäume — z.B. im Gebiet des Kodierens von Graphen — sind ebenfalls in [1] zu finden.

## Literatur

- [1] Y.-T. CHIANG, C.-C. LIN, AND H.-I. LU, *Orderly spanning trees with applications to graph encoding and graph drawing*, in Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Washington, D.C, USA, 2001, ACM Press, pp. 506–515. Eine überarbeitete und erweiterte Version kann unter folgender URL gefunden werden: <http://arxiv.org/pdf/cs/0102006>.
- [2] R. C.-N. CHUANG, A. GARG, X. HE, M.-Y. KAO, AND H.-I. LU, *Compact encodings of planar graphs via canonical ordering and multiple parentheses*, in Proceedings of the 25th International Colloquium on Automata, Languages, and Programming (ICALP), K. G. Larsen, S. Skyum, and G. Winskel, eds., vol. 1443 of Lecture Notes in Computer Science, Aalborg, Denmark, 1998, Springer-Verlag, pp. 118–129.
- [3] U. FÖSSMEIER, G. KANT, AND M. KAUFMANN, *2-visibility drawings of planar graphs*, in Proceedings of the 4th International Symposium on Graph Drawing (GD), S. North, ed., vol. 1190 of Lecture Notes in Computer Science, California, USA, 1996, Springer-Verlag, pp. 155–168.
- [4] J. HOPRCOFT AND R. E. TARJAN, *Efficient planarity testing*, Journal of the ACM (JACM), 21(4) (1974), pp. 549 – 568.
- [5] Seminar: Zeichnen von Graphen, WS 2004/05, Universität Konstanz. <http://www.inf.uni-konstanz.de/algo/lehre/ws04/gd/>.