Algorithmen zur Visualisierung von Graphen

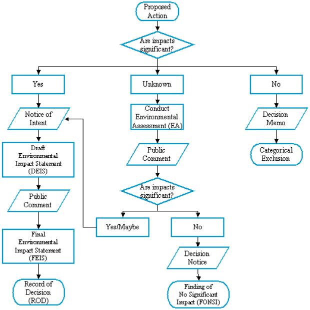
Symmetrien in Serienparallelen Graphen
Geradlinige planare Gitterzeichnungen

Vorlesung im Sommersemester 2009

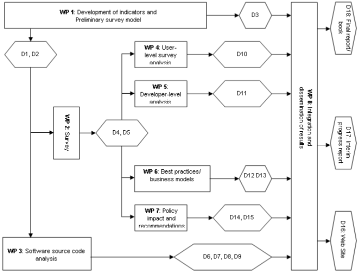
Martin Nöllenburg

23.07.2009

SP-Graphen in Anwendungen



Ablaufdiagramme



PERT-Diagramme

(Program Evaluation and Review Technique)

Außerdem: Linearzeitalgorithmen für sonst NP-vollständige Probleme (z.B. Maximum Independent Set)

Symmetrien in SP-Zeichnungen

Definition: Automorphismen eines DAG

Ein Automorphismus eines DAG $G = (V, E)$ ist eine Knotenpermutation $\pi : V \to V$, die Adjazenzen respektiert und alle Kantenrichtungen erhält oder alle Kantenrichtungen umdreht:

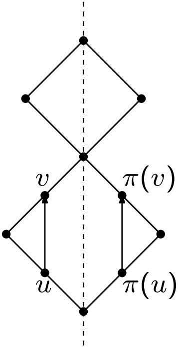
$$(u, v) \in E \Leftrightarrow (\pi(u), \pi(v)) \in E$$

oder

$$(u, v) \in E \Leftrightarrow (\pi(v), \pi(u)) \in E.$$

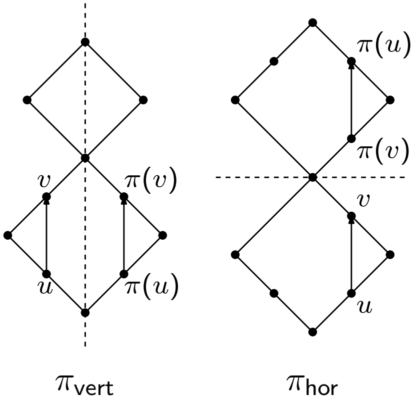
Die Automorphismen von G bilden mit der Hintereinanderausführung eine Gruppe.

Symmetrien in SP-Graphen

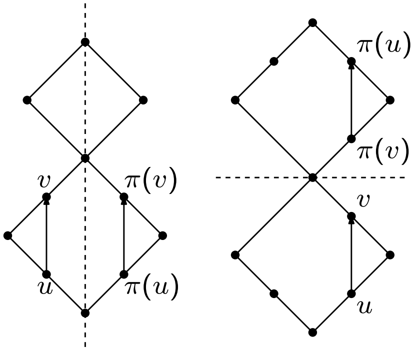


π_{vert}

Symmetrien in SP-Graphen

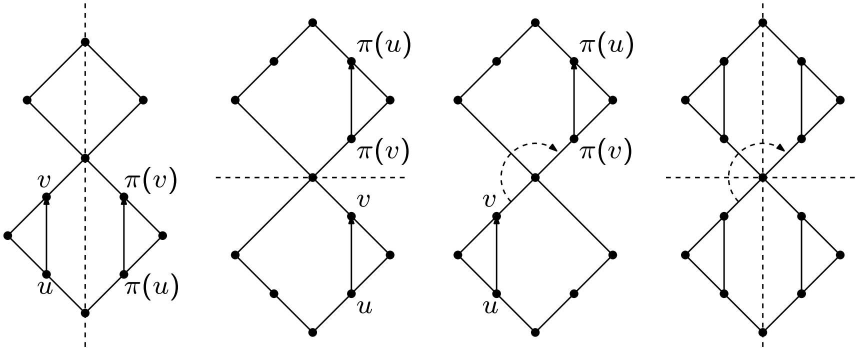


Symmetrien in SP-Graphen



π_{vert} π_{hor} π_{rot}

Symmetrien in SP-Graphen



π_{vert} π_{hor} π_{rot} $\{\pi_{\text{vert}}, \pi_{\text{hor}}, \pi_{\text{rot}}\}$

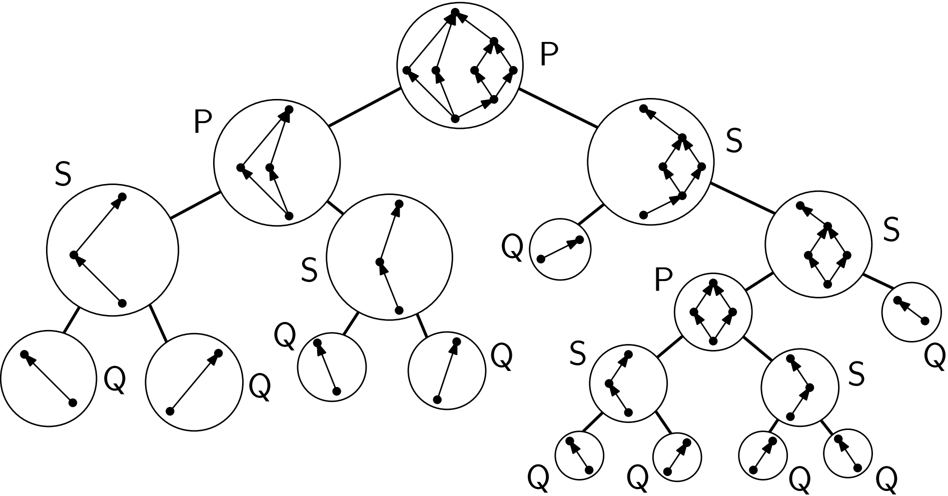
Symmetrien in SP-Zeichnungen

Satz (Hong, Eades, Lee ’00)

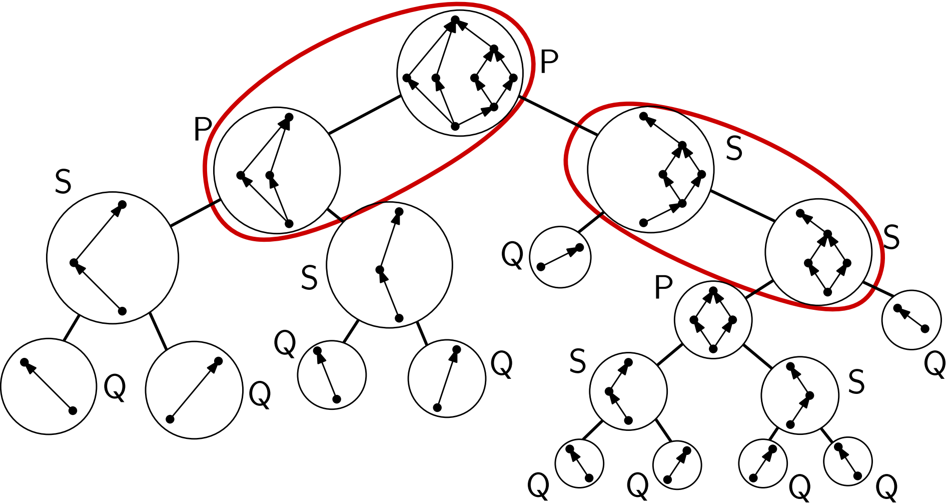
Die in einem kreuzungsfreien Aufwärtslayout eines SP-Graphen darstellbaren Symmetrien sind entweder

1. $\{\text{id}\}$
2. $\{\text{id}, \pi\}$ mit $\pi \in \{\pi_{\text{vert}}, \pi_{\text{hor}}, \pi_{\text{rot}}\}$
3. $\{\text{id}, \pi_{\text{vert}}, \pi_{\text{hor}}, \pi_{\text{rot}}\}$.

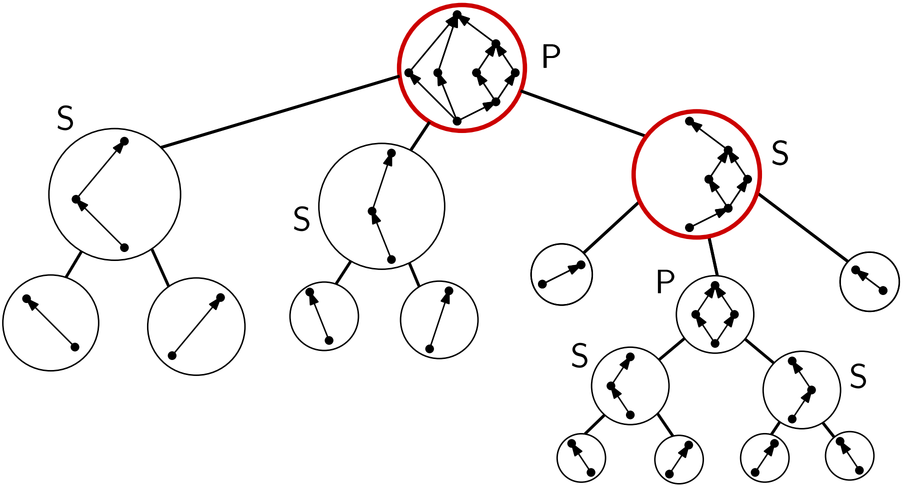
Knotenkodierung im Dekompositionsbaum



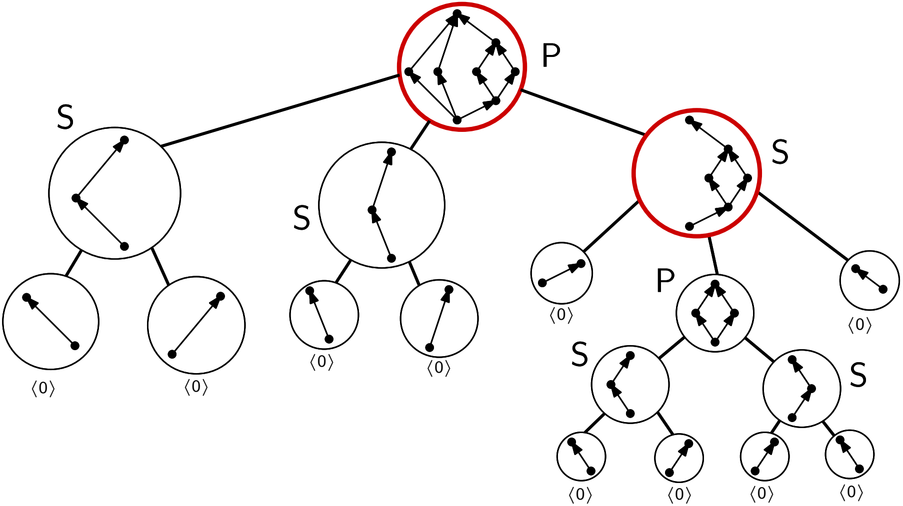
Knotenkodierung im Dekompositionsbaum



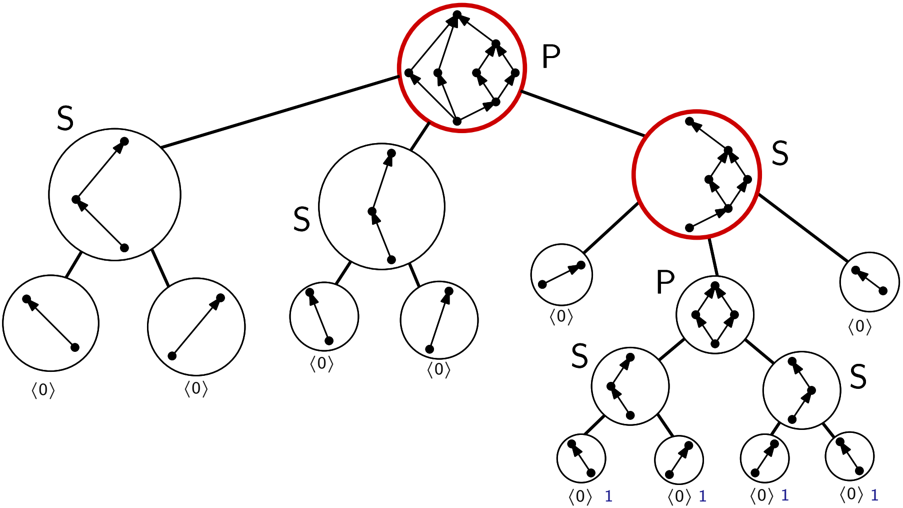
Knotenkodierung im Dekompositionsbaum



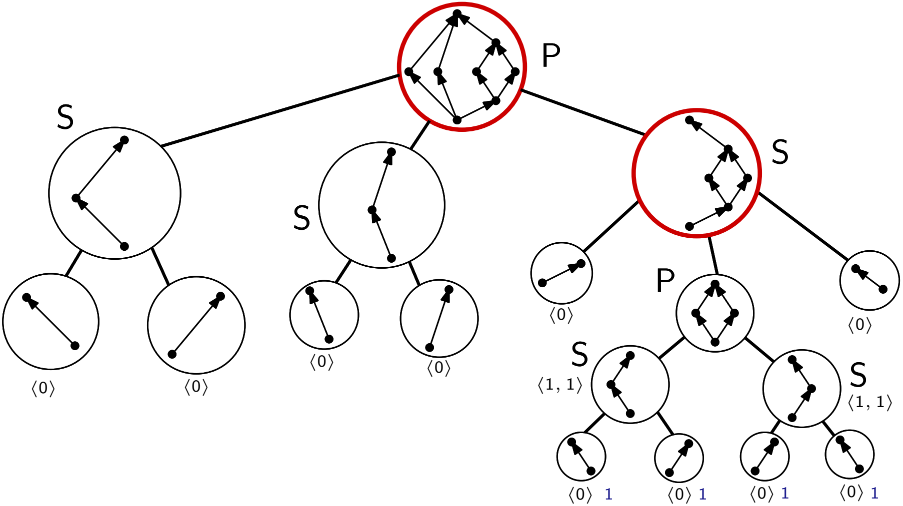
Knotenkodierung im Dekompositionsbaum



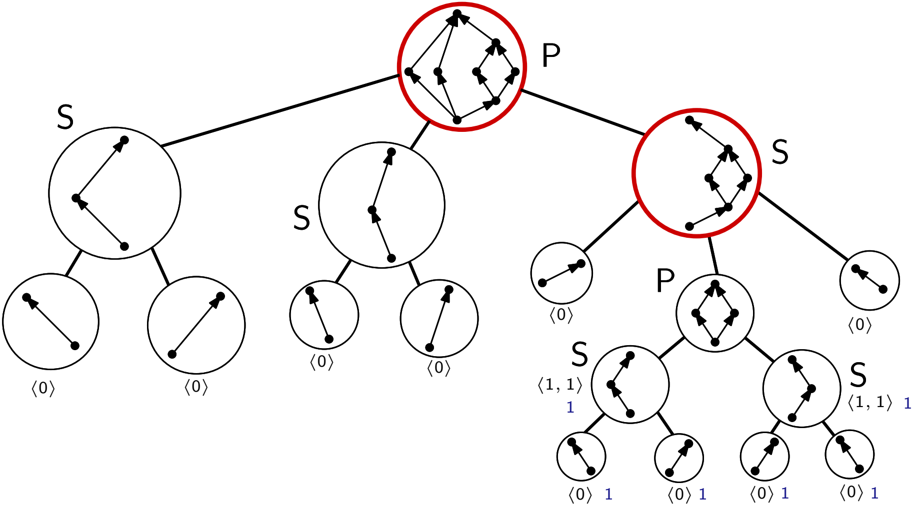
Knotenkodierung im Dekompositionsbaum



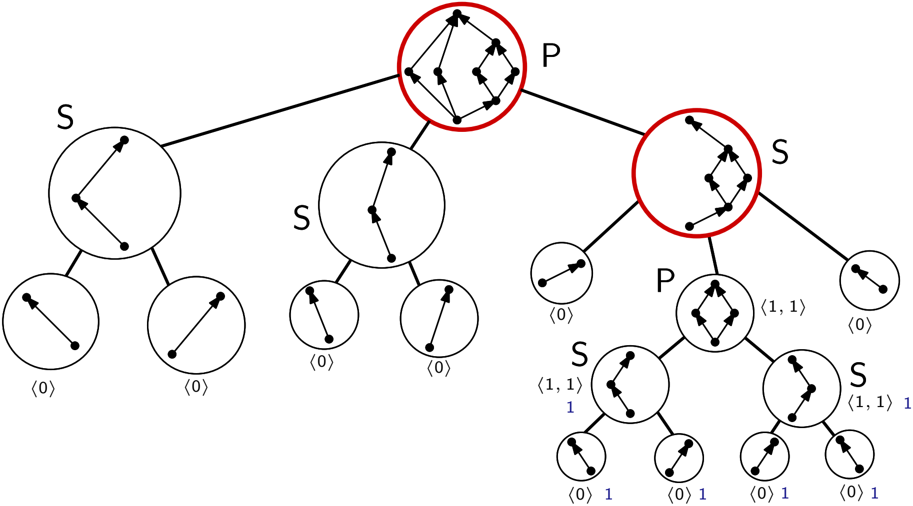
Knotenkodierung im Dekompositionsbaum



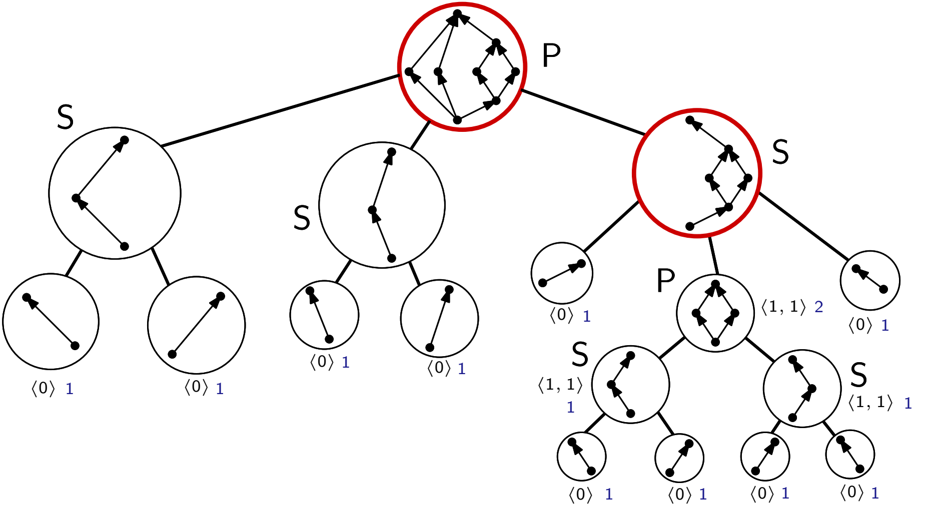
Knotenkodierung im Dekompositionsbaum



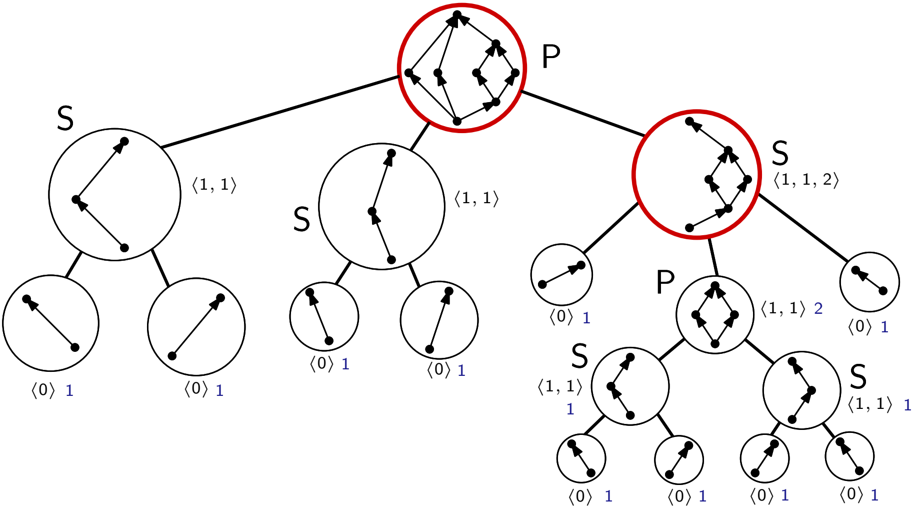
Knotenkodierung im Dekompositionsbaum



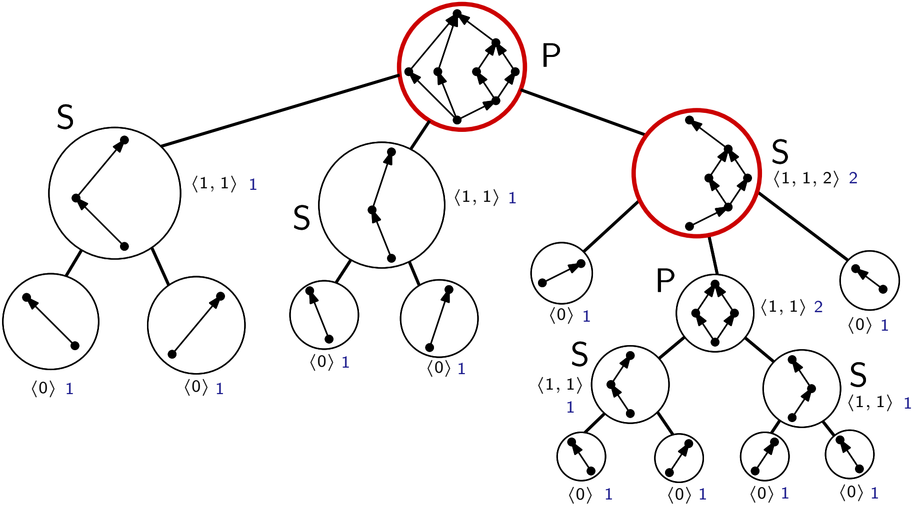
Knotenkodierung im Dekompositionsbaum



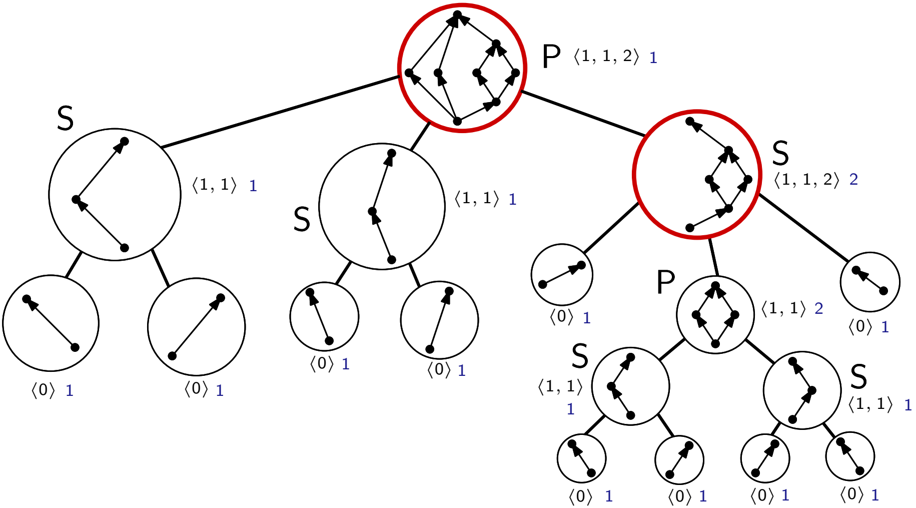
Knotenkodierung im Dekompositionsbaum



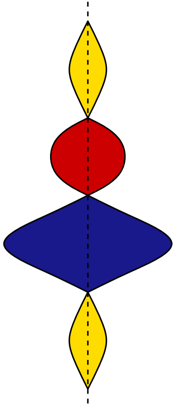
Knotenkodierung im Dekompositionsbaum



Knotenkodierung im Dekompositionsbaum

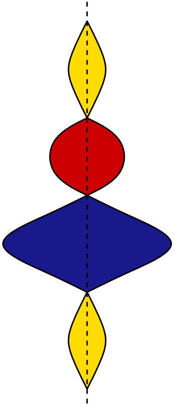


Beweisidee vertikale Symmetrie

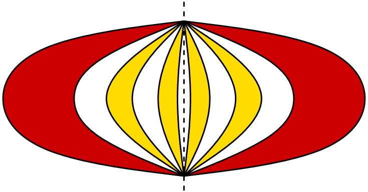


S-Knoten

Beweisidee vertikale Symmetrie

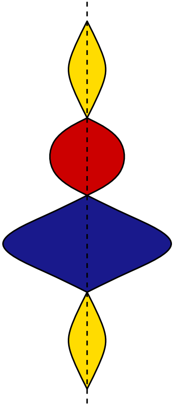


S-Knoten

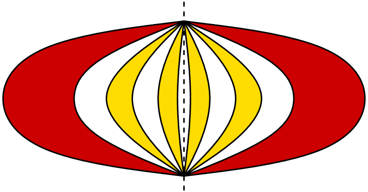


P-Knoten, alle Klassen gerade Anzahl

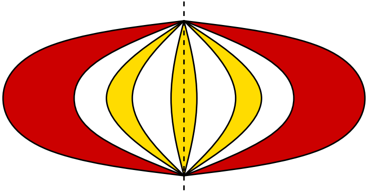
Beweisidee vertikale Symmetrie



S-Knoten
alle Knoten v-symm.

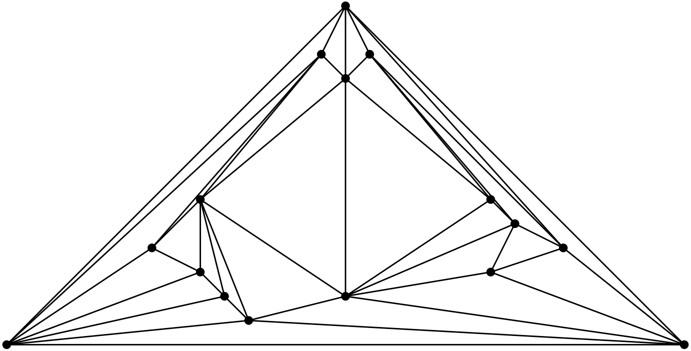


P-Knoten, alle Klassen gerade Anzahl



P-Knoten, eine Klasse ungerade Anzahl & v-symm.

Inkrementelles Gitterlayout für planare Graphen



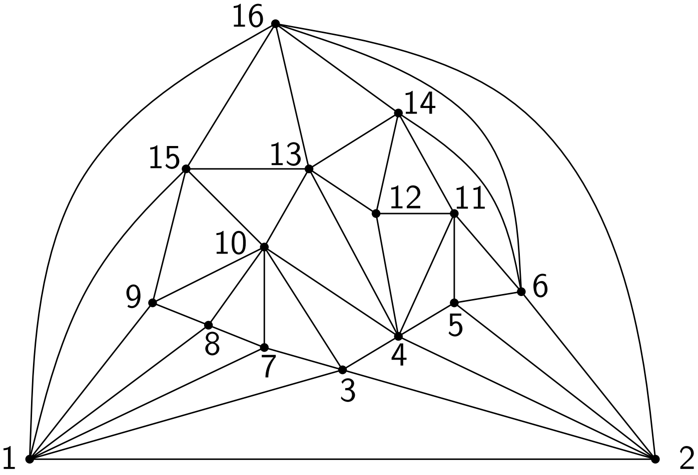
Kanonische Knotenordnung

Definition: Kanonische Knotenordnung

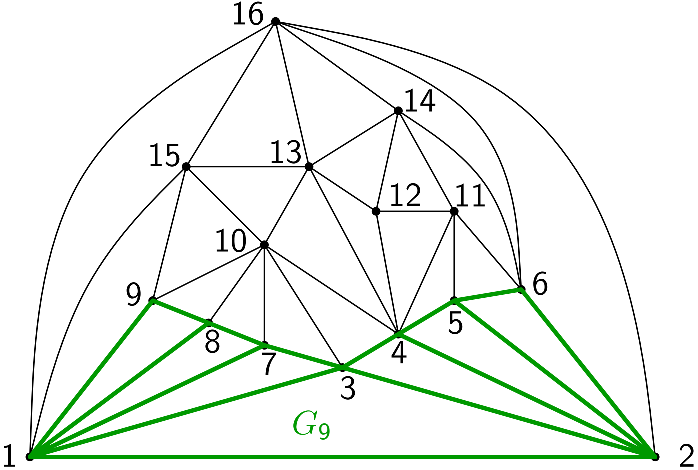
Sei $G = (V, E)$ ein triangulierter, planar eingebetteter Graph mit $n \geq 3$ Knoten. Eine Knotenordnung $\pi = (v_1, v_2, \dots, v_n)$ heißt *kanonische Ordnung*, falls gilt

1. der von $\{v_1, \dots v_k\}$ induzierte Teilgraph G_k ist 2-zusammenhängend und intern trianguliert
2. (v_1, v_2) ist Außenkante von G_k
3. für $k < n$ liegt v_{k+1} in der äußeren Facette von G_k und alle Nachbarn von v_{k+1} in G_k bilden ein Intervall auf dem Rand $C_o(G_k)$ der äußeren Facette von G_k

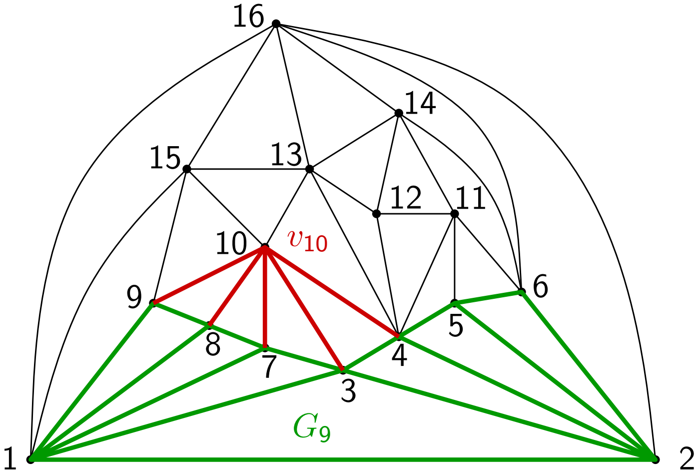
Beispiel kanonische Ordnung



Beispiel kanonische Ordnung



Beispiel kanonische Ordnung

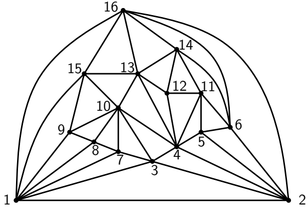


Algorithmus kanonische Ordnung

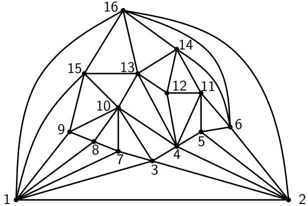
Algorithm 1: Canonical-Ordering

forall $v \in V$ **do**
  chords(v) $\leftarrow$ 0; out(v) $\leftarrow$ false; mark(v) $\leftarrow$ false;
out(v_1), out(v_2), out(v_n) $\leftarrow$ true;
for $k = n$ **to** 3 **do**
  wähle $v \neq v_1, v_2$ mit mark(v) $=$ false, out(v) $=$ true, chords(v) $=$ 0;
  $v_k \leftarrow v$; mark(v) $\leftarrow$ true;
  $(w_1 = v_1, w_2, \ldots, w_{t-1}, w_t = v_2) \leftarrow C_o(G_{k-1})$;
  $(w_p, \ldots, w_q) \leftarrow$ unmarkierte Nachbarn von v_k;
  out(w_i) $\leftarrow$ true for all $p < i < q$;
  aktualisiere chords($\cdot$) für diese w_i und ihre Nachbarn;

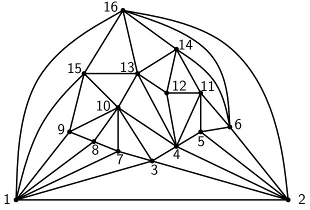
Beispiel

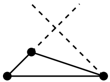


Beispiel

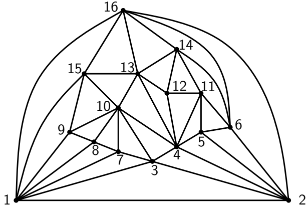


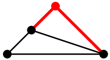
Beispiel



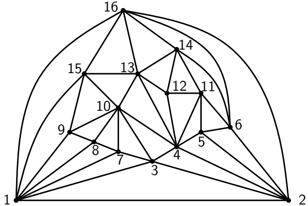


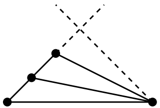
Beispiel



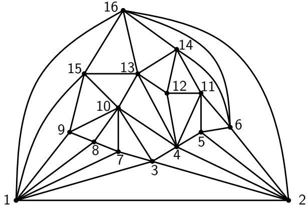


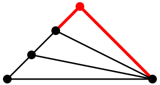
Beispiel



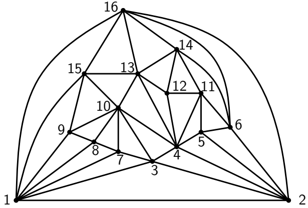


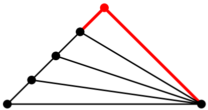
Beispiel



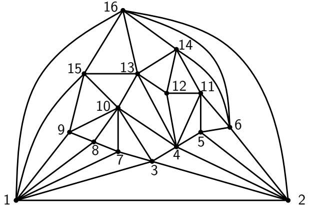


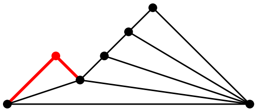
Beispiel



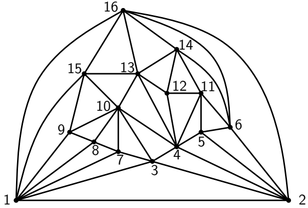


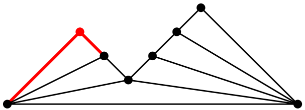
Beispiel



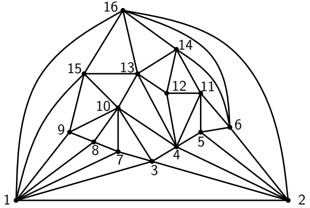


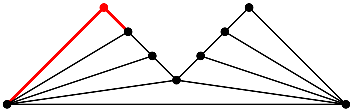
Beispiel



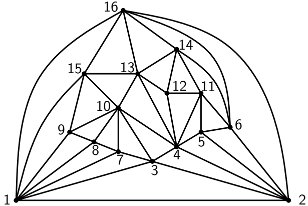


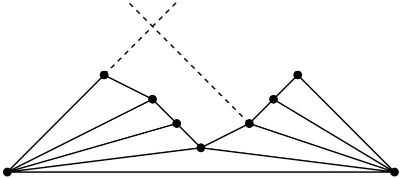
Beispiel



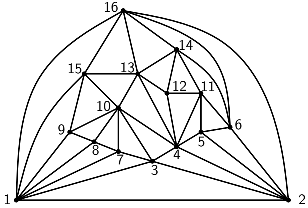


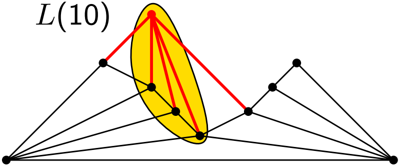
Beispiel



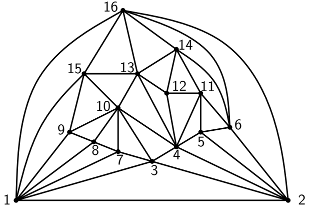


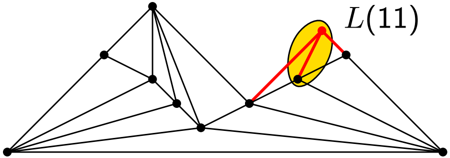
Beispiel



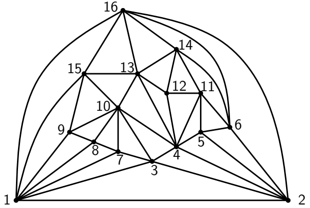


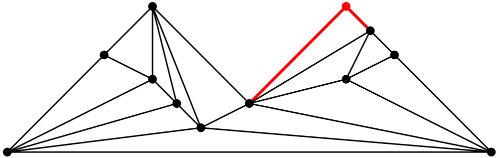
Beispiel



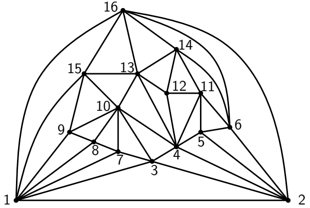


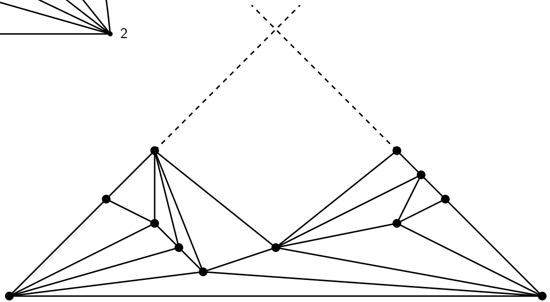
Beispiel



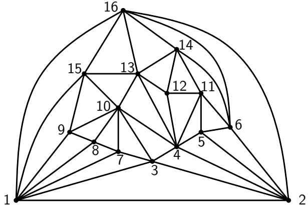


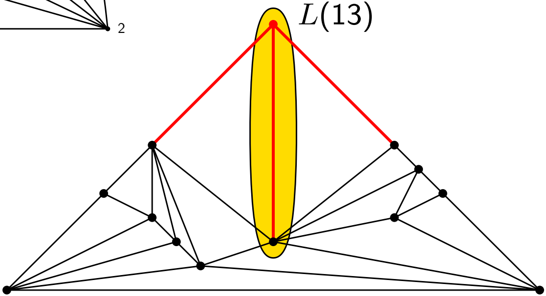
Beispiel



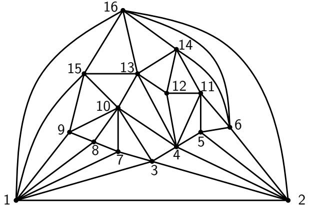


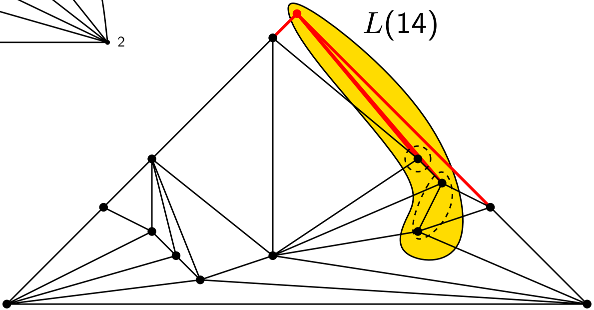
Beispiel



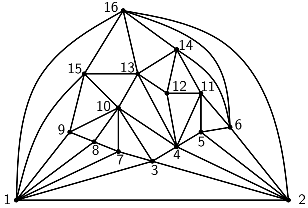


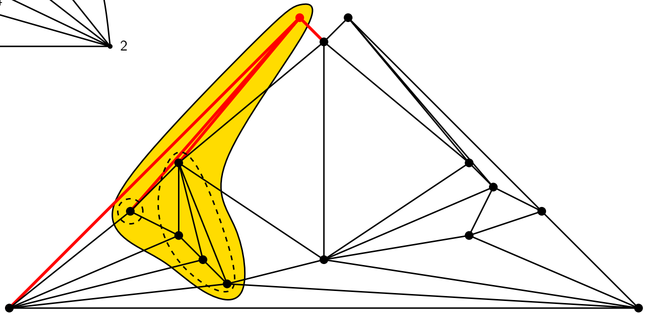
Beispiel



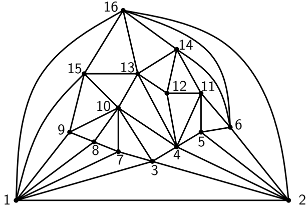


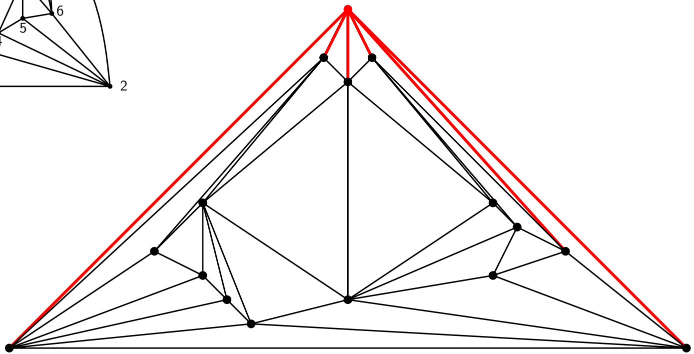
Beispiel



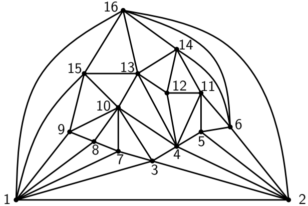


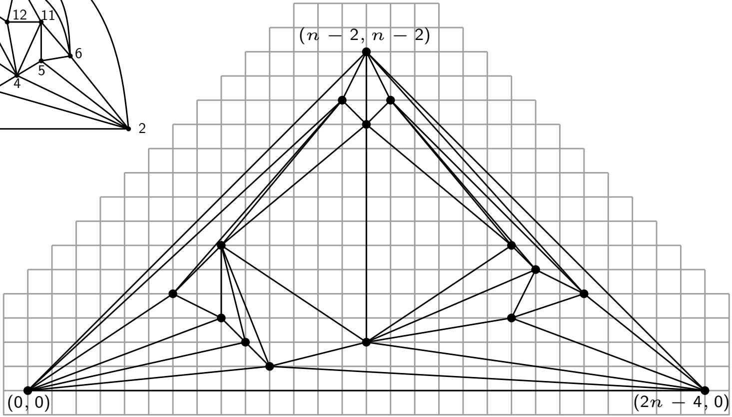
Beispiel





Beispiel





Wie geht es weiter?

- Praktikum Graphenzeichnen im WS 09/10
- Seminar Algorithmische Geometrie im WS 09/10
- Studien- und Diplomarbeiten, insbesondere im Bereich schematisches Graphenzeichnen

Wie geht es weiter?

- Praktikum Graphenzeichnen im WS 09/10
- Seminar Algorithmische Geometrie im WS 09/10
- Studien- und Diplomarbeiten, insbesondere im Bereich schematisches Graphenzeichnen
- Sprechstunden in den Semesterferien:
  - bis 17. 08.
  - notfalls 26. 08.
  - 17. – 18. 09.
  - ab 28. 09.

Wie geht es weiter?

- Praktikum Graphenzeichnen im WS 09/10
- Seminar Algorithmische Geometrie im WS 09/10
- Studien- und Diplomarbeiten, insbesondere im Bereich schematisches Graphenzeichnen
- Sprechstunden in den Semesterferien:
  - bis 17. 08.
  - notfalls 26. 08.
  - 17. – 18. 09.
  - ab 28. 09.

Viel Erfolg in der Prüfung!