Karlsruher Institut für Technologie
Fakultät für Informatik
ITI Wagner

Algorithmen zur Visualisierung von Graphen
WS 2016/17
Tamara Mchedlidze, Benjamin Niedermann

Viertes Übungsblatt

Ausgabe: 16. Dezember 2016
Besprechung: 11. Januar 2017

1 Feedback Arc Set

In der Vorlesung wurden die beiden folgenden Probleme Minimum Feedback Arc Set und Minimum Feedback Set vorgestellt.

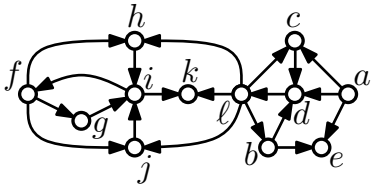
Problem 1.1 (Minimum Feedback Arc Set)**.** *Sei $D = (V, A)$ ein gerichteter Graph. Bestimme eine Menge $A_f \subset A$ minimaler Kardinalität, so dass $D_f = (V, A \setminus A_f)$ azyklisch ist.*

Problem 1.2 (Minimum Feedback Set)**.** *Sei $D = (V, A)$ ein gerichteter Graph. Bestimme eine Menge $A_r \subset A$ minimaler Kardinalität, so dass $D_r = (V, A \setminus A_r \cup \mathrm{rev}(A_r))$ azyklisch ist. Dabei bezeichne* $\mathrm{rev}(A)$ *die Kantenmenge, die man erhält wenn die Richtung jeder Kante der Menge A invertiert wird.*

Wir haben gesehen, dass jedes Feedback Set ein Feedback Arc Set ist, umgekehrt gilt dies im Allgemeinen jedoch nicht. Für minimale Mengen gilt die Äquivalenz: Zeigen Sie, dass die Lösungsmengen der beiden Probleme 1.1 und 1.2 identisch sind.

2 Lagenlayout

Führen Sie die einzelnen Schritte des Sugiyama-Frameworks zur Generierung von Lagenlayouts exemplarisch für den nebenstehenden Graphen aus. Entfernen Sie dazu gerichtete Kreise indem Sie für eine möglichst kleine Menge an Kanten die Richtung umkehren, finden Sie eine Lagenzuordnung minimaler Höhe bei maximaler Breite 4 und ordnen sie die Knoten innerhalb der Lagen so an, dass die Anzahl an Kreuzungen minimal ist.
Hinweis: Bestimmen Sie dazu in jedem Schritt manuell eine optimale Lösung des jeweiligen Teilproblems.



3 Kreuzungen bei Lagenlayouts

Zeigen Sie, dass die Baryzenter-Heuristik zur einseitigen Kreuzungsreduktion die optimale Lösung liefert, falls diese keine Kreuzung enthält.

4 Fehlstände Zählen

(a) Sei $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ eine Permutation. Ein Paar (i, j) mit $1 \leq i < j \leq n$ heißt Inversion, wenn $\pi(i) > \pi(j)$. Entwerfen Sie einen Algorithmus, der die Anzahl der Inversionen einer Permutation von n Elementen in $O(n \log n)$ Zeit berechnet.
Hinweis: Denken Sie an Sortieralgorithmen wie Mergesort.

(b) Gegeben sei ein einfacher, bipartiter Graph $G = (V, E)$, dessen Knoten gemäß der Bipartition auf zwei parallele Geraden verteilt sind. Die Knoten seien disjunkt und die Kanten geradlinig gezeichnet. Entwerfen Sie einen Algorithmus, der die Anzahl der Kreuzungen in einer Laufzeit $O(|E| \log |V|)$ bestimmt. Begründen Sie, warum es nicht möglich ist, in dieser worst-case-Laufzeit alle Kreuzungen (d.h. die Paare betroffener Kanten) *auszugeben*.