

Blatt2

Tobias

19.5.2021

Aufgabe 2.1

Graph 1)

a) Ordnung = 5 %Anzahl der Knoten

b)

```
0 1 0 1 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 1
0 0 0 0 0
```

c)

Quellen = A (Valenzen = 2)
Senken = C (1), E (1)

e) gerichtet azyklisch
besitzt definierte richtungen aber keine schleifen/zykeln

f)

+verbindung zwischen AC und AE
siehe Bild

Graph 2)

a) Ordnung = 5

b)

```
0 1 0 0 0
0 0 1 0 0
0 0 0 1 1
```

```
1 0 0 0 0
1 0 0 0 0
```

c)

Keine Quellen oder Senken vorhanden, da zyklisch.

Graph 3)

a) Ordnung = 5

b)

```
0 1 0 0 0
0 0 1 0 0
0 1 1 1 0
0 0 0 0 1
0 0 0 0 0
```

c)

Quellen = A (1)
Senken = E (1)

Graph 4)

a) Ordnung = 5

b)

```
0 1 1 0 0
1 0 1 0 0
1 1 0 1 1
0 0 1 0 1
0 0 1 1 0
```

Graph 5)

a) Ordnung = 7

b)

```
0 1 0 1 0 0 0
1 0 1 0 0 0 0
0 1 0 1 0 0 0
1 0 1 0 1 0 0
0 0 0 1 0 1 1
```

0 0 0 0 1 0 0
0 0 0 0 1 0 0

d) Nachbarn $G = E$

Graph 6)

a) Ordnung = 7

b) 0 0.5 0 0 0 0 0
0 0 0.25 0 0 0 0
0 0 0 0.15 0 0 0
0.75 0 0 0 0 0 0
0 0 0 0 0 0.35 0.45
0 0 0 0 0 0 0
0 0 0.65 0.85 0 0 0

c) Quellen = E (2)
Senken = F (2)

d) Nachbarn $G = F$ (C,D,E,F)

Aufgabe 2.2

siehe Blatt

Aufgabe 2.3

a) Typen von Anforderungen zum Zeichnen von Graphen:

1. Auflösungsbezogene Anforderungen

- untere Schranke für Knotenabstand
- untere Schranke für den Winkel zwischen zwei aufeinanderfolgenden inzidenten Kanten
- untere Schranke für den Abstand zwischen Knoten und inzidenten Kanten

2. Ästhetikbezogene Anforderungen

- Kantenschnitte. Minimierung der Anzahl der Schnittpunkte von Kanten.
- Fläche. Graph wird auf möglichst kleiner Fläche gezeichnet. Die Fläche wird z.B. durch eine BoundingBox oder die konvexe Hülle gemessen.
- Gesamtkantenlänge. Gesamtkantenlänge wird minimiert. Dies ist nur bei skalierungsunabhängigen Zeichnungen sinnvoll.
- Maximale Kantenlänge. Länge der längsten Kante wird minimiert. Auch dies ist nur bei skalierungsunabhängigen Zeichnungen sinnvoll.
- Gleichförmige Kantenlänge. Varianz der Kantenlängen wird minimiert.
- Gesamtanzahl Knicke. Zahl aller Knicke aller Kanten soll minimiert werden. Dies ist besonders bei orthogonalem Zeichnen sinnvoll.

- Maximale Knickanzahl. Maximum von Knicken einer Kante wird minimiert.
- Gleichmäßige Knickanzahl. Varianz der Knickanzahl wird minimiert.
- Winkelauflösung. Kleinsten auftretenden Winkel zwischen zwei Kanten maximieren
- Seitenverhältnis. Seitenverhältnis des kleinsten umschriebenen Rechtecks soll nahe 1 sein.
- Symmetrie. Symmetrien des Graphen sollen möglichst gut abgebildet werden

3. Anwendungsbezogene Anforderungen

- Zentrum. Ein gegebener Knoten wird im Zentrum der Zeichnung fixiert.
- Außen. Ein bestimmter Knoten wird der äußeren Facette benachbart.
- Cluster. Eine gegebene Menge von Knoten wird nah beieinander angeordnet.
- Vertikal-Horizontal-Folge. Ein vorgegebener Weg wird vertikal/horizontal gezeichnet.
- Form. Zeichne einen gegebenen Teilgraphen mit vorgegebener Form

b) Zeichenstrategien

- Topologie-Form-Metrik-Ansatz
 - Topologie, Form und Metrik müssen gleich bleiben.
 - Vereinfachung durch Planarisierung (keine Kantenschnitte), Orthogonalisierung (+ Minimierung der Knicke) und Kompaktifizierung (Fläche minimiert).
- Hierarchischer Ansatz (azyklisch) Aufgabe 2.4
 - Jeder Knoten auf eine Schicht (+ Dummy-Knoten)
 - Ordnung innerhalb der Schichten + Kantenschnitte minimieren
 - Dummy-Knoten durch Knicke ersetzt (minimierung) + Symmetrien betonen
- Sichtbarkeitsansatz
 - Planarisierung (Topologie-Form-Metrik Ansatz)
 - Sichtbarkeitsschritt (Knoten horizontal, Kanten vertikal) (Erreichbare Punkte sind sichtbar.)
 - Ersetzungsschritt (Knoten durch Punkt und vertikale Kante durch Polylinie)
- Verfeinerungsansatz
 - Planarisierung
 - Vereinfachung (Fügt Knoten hinzu damit Facette immer drei Seiten hat)
 - Trieangulierungszeichnung (Zeichnung und Entfernung der Dummy-Knoten)

c) Topologie

Graph 1 & 2 gleichen sich, da die gleichen Knoten miteinander verbunden sind. Der Graph 2 wird zu Graph 1 lediglich an zwei Stellen rotiert und neu verbunden. (gleiche Metrik) (Deformation ohne Vertauschung der Facetten)

Aufgabe 2.4

Algorithmus von Sugiyama

siehe Blatt

Der Kode funktioniert nur wenn keine Zykeln vorhanden sind. Da der Schritt eigentlich dafür gemacht ist, #Zykeln zu entfernen, ist er nicht wirklich Sinnvoll.

```
f = function(W,K,S,Ks){
  for (i in Ks){
```

```

b = W == i
a = 0
if (sum(iffelse(b == FALSE, a, a+1))<1){
  S = c(i,S)
}
}
if (length(W)==0){return(c(S))}
else{
  loeschen = which(K %in% c(S))
  Ks = Ks[!Ks %in% S]
  K = K[-c(loeschen)]
  W = W[-c(loeschen)]
  f(W,K,S,Ks)}
}

KantenWurzel = {c(1,1,2,3,3,3,4,4,4,4,6 ,7 ,7 ,8 ,9,10,11)}
KantenKinder = {c(3,4,6,2,7,8,5,6,8,9,10,10,11,7,11,12,12)}
# Senke wenn Knoten keine Wurzel
Senken = c()
Ks = c(1:12)

Senken = f(KantenWurzel,KantenKinder,Senken,Ks)
Senken

## [1] 1 4 3 8 2 9 7 6 11 10 12 5

```

Aufgabe 2.5

inorder: d c e b g f a h j i k
preorder: a b c d e f g h i j k
Postorder: d e c g f b j k i h a