Karlsruher Institut für Technologie
Fakultät für Informatik
ITI Wagner

Algorithmische Kartografie
Sommersemester 2013
Martin Nöllenburg/Benjamin Niedermann

Musterlösung für Übungsblatt 2

Ausgabe: Dienstag, 30. April 2013
Abgabe: Bis spätestens Dienstag, 07. Mai 2013 um 12:00 Uhr.

Hinweis: Abgabe ist sowohl in den Vorlesungen und Übungen als auch im Raum 322 des Informatik-Hauptgebäudes möglich.

1 Einstieg

In der Vorlesung vom 30.04. wurde ein Vorgehen zur Schematisierung von Karten vorgestellt. Skizzieren Sie kurz dieses Verfahren und begründen Sie die Korrektheit.

Lösung: Gegeben sei eine Karte M, die schematisiert werden soll.

1. Überprüfe in $O(n \log n)$ Zeit, ob M eine äquivalente, monotone Karte M' besitzt. Falls nicht, breche das Verfahren ab. Dies ist ausschließlich ein Existenztest, M' wird nicht erstellt. Allerdings erhält man eine partielle Ordnung der Kurven von M'.
2. Erweitere in $O(n \log n)$ Zeit die partielle Ordnung zu einer totalen Ordnung. Nach Lemma 2 ist jede monotone Karte M'' mit der selben Ordnung äquivalent zu M' und somit zu M.
3. Konstruiere die Karte M'' (siehe Aufgabe 3).

2 Monotone Karten

Betrachten Sie eine monotone Karte M, d.h. eine Menge an einfachen, x-monotonen Kurven, die ausschließlich in Endpunkten übereinstimmen dürfen.

1. Zeigen Sie, dass die Vertikalordnung $\succ$ für eine monotone Karte azyklisch ist.
2. Ist sie auch transitiv?
3. Geben Sie einen Algorithmus an, der die Vertikalordnung $\succ$ zu einer totalen Ordnung erweitert. Welche Laufzeit besitzt Ihr Algorithmus?
   Hinweis: Laut Vorlesung gibt es einen Algorithmus, der diese totale Ordnung in $O(n \log n)$ Zeit berechnet.

Lösung:

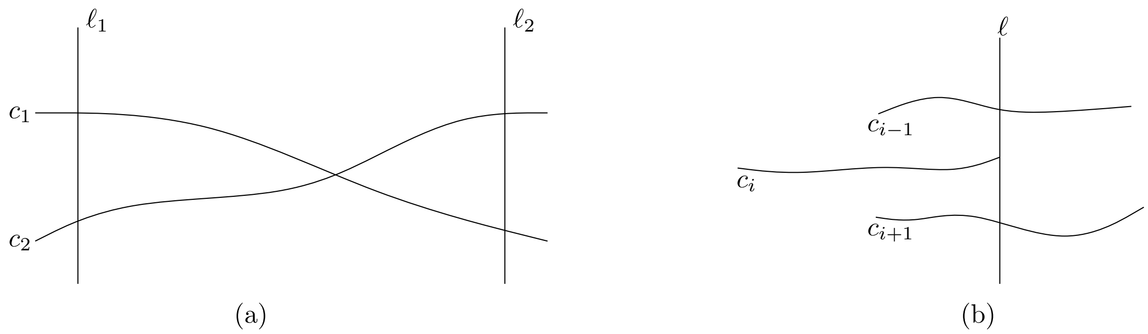

Abbildung 1: a) ℓ_1 und ℓ_2 begrenzen den Bereich in dem ein Ordnungswechsel passieren muss. b) Der rechte Endpunkt von c_i liegt am weitesten rechts unter den rechten Endpunkten aller Kurven.

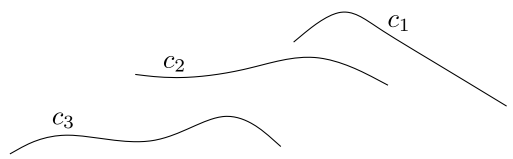

Abbildung 2: c_1 liegt oberhalb von c_2 und c_2 liegt oberhalb von c_3, aber c_1 liegt nicht oberhalb von c_3.

1. Annahme: Es gibt monotone Karte M, für welche die Aussage nicht gilt. Dann gibt es k Kurven $c_1, \ldots, c_k$, sodass $c_1 > \ldots > c_k > c_1$. Sei k minimal gewählt.

   Fall $k = 2$: Wegen $c_1 > c_2$ gibt es eine vertikale Gerade ℓ_1, die zuerst c_1 und dann c_2 schneidet, wenn sie von oben nach unten orientiert ist. Analog, wegen $c_2 > c_1$ gibt es eine Gerade ℓ_2, die zuerst c_2 und dann c_1 schneidet, wenn sie von oben nach unten orientiert ist. O.B.d.A. sei ℓ_1 links von ℓ_2. Wegen der x-Monotonie beider Kurven müssen sich c_1 und c_2 in dem Bereich zwischen ℓ_1 und ℓ_2 schneiden, um einen Ordnungswechsel zu erzwingen. Dies ist ein Widerspruch zur Kreuzungsfreiheit der Kurven; siehe Abbildung 1a.

   Fall $k > 2$: Sei c_i mit $1 \leq i \leq k$ die Kurve, deren rechter Endpunkt (x_i, y_i) am weitesten links liegt. Hinzu sei ℓ die vertikale Gerade durch (x_i, y_i); siehe Abbildung 1b. Die Gerade ℓ schneidet zuerst c_{i-1}, dann c_i und schließlich c_{i+1}, wenn sie von oben nach unten orientiert ist (für $i = k$ setze $c_{i+1} = c_1$ und für $i = 1$ setze $c_{i-1} = c_k$). Dass ℓ die Kurven c_{i-1} und c_{i+1} schneidet leitet sich aus $c_{i-1} > c_i > c_{i+1}$ und der Wahl von c_i ab. Die Reihenfolge ergibt sich aus der x-Monotonie der Kurven. Dann gilt aber auch $c_1 > \cdots > c_{i-1} > c_{i+1} > \cdots > c_k$, was ein Widerspruch zur Minimalität von k ist.

2. Die Vertikalordnung ist nicht transitiv, siehe Abbildung 2.

3. Sei $M = \{c_1, \ldots, c_n\}$ die gegebene Karte. Aus technischen Gründen fügen wir die Strecke c_0 mit Endpunkten $(-\infty, \infty)$ und (∞, ∞) hinzu. Erstelle einen Baum T mit Knoten $c_0, \ldots, c_n$, so dass c_i genau dann Kind von c_j ist, wenn c_j direkt oberhalb des rechten Endpunkt von c_i liegt (siehe Abbildung 3). Insbesondere ist c_0 die Wurzel des Baums.

   Der Baum T kann mithilfe eines Sweep-Line-Verfahrens in $O(n \log n)$ Zeit bestimmt werden: Sei hierzu ℓ eine vertikale Sweep-Line, die von links nach rechts läuft und die Anfangs- und Endpunkte der Kurven als Events abarbeitet. Der *Sweep-Line-Zustand* S wird repräsentiert durch einen binären Baum und speichert die Kurven, die ℓ an der aktuellen Position von oben nach unten schneiden. Ein Kurve c_i wird in S aufgenommen, wenn deren linker Endpunkt abgearbeitet wird. Sie wird aus S entfernt, wenn deren rechter Endpunkt

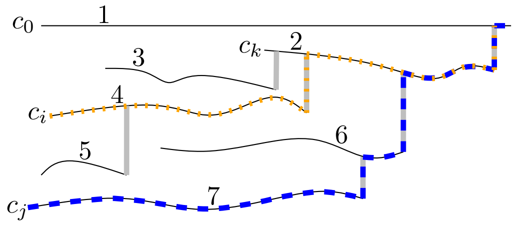

Abbildung 3: Illustration der totalen Ordnung von sieben Kurven. Die grauen vertikalen Segmente entsprechen den Kanten im Baum T. Die Nummerierung der Kurven entspricht einer Inorder-Traversierung der Knoten. Der rechts-gerichtete Pfad von c_i ist orange markiert und der rechts-gerichtete Pfad von c_j ist blau markiert. Die Pfade vereinigen sich auf c_k.

abgearbeitet wird. Hinzu wird in diesem Fall c_i als Kind der Kurve gesetzt, welche die Sweep-Line direkt über c_i schneidet. Für Illustration siehe Folien der Übung.

Man benötigt $O(n \log n)$ Zeit um Anfang- und Endpunkte (Events) zu sortieren. Jedes Event wird einmal in S in $O(\log n)$ Zeit eingefügt und einmal entfernt. Es ergibt sich damit ein Gesamtaufwand von $O(n \log n)$.

Der so erzeugte Baum T kann nun dazu verwendet werden, um eine totale Ordnung zu erzeugen, welche die *Oberhalb*-Relation einhält. Hierzu traversiere T nach dem Inorder-Prinzip und Nummeriere die Knoten gemäß ihres Auftretens (siehe Folien der Übung).

Beweis, dass diese totale Ordnung die *Oberhalb*-Relation erhält: Verbinde mithilfe eines vertikalen Segment den rechten Endpunkt einer Kurve mit der Kurve, die direkt oberhalb dieser liegt (entspricht den Kanten von T). Ein *rechts-gerichteter Pfad* einer Kurve c_i erhält man, indem man beginnend am linken Startpunkt von c_i zum rechten Endpunkt von c_0 entlang der Kurven und vertikalen Segmente geht und dabei nie nach links geht (siehe Abbildung 3).

Die rechts-gerichteten Pfade zweier Kurven c_i und c_j können sich nicht schneiden, da wenn sie sich treffen, dann entlang einer Kurve, sodass sie sich zu einem Pfad vereinen.

Sei c_k die Kurve, an der sich c_i und c_j treffen. Sei c_i oberhalb von c_j, dann gilt entweder $c_i = c_k$ oder c_i und c_j befinden sich in unterschiedlichen Teilbäumen T_i bzw. T_j von c_k. Im ersten Fall ist c_i der Vater von c_j, sodass bei einer Inorder-Traversierung c_i vor c_j nummeriert wird. Im anderen Fall ist T_i links von T_j an c_k angehängt. Damit ergibt sich, dass c_i vor c_j nummeriert wird.

3 Kleinere Aufgaben

1. Zeigen oder widerlegen Sie: Lemma 2 gilt auch für nicht monotone Kurven.

   Erinnerung Lemma 2: *Zwei monotone Karten M und M' sind äquivalent gdw. sie die gleiche Vertikalordnung $\succ$ definieren.*

2. Zeigen Sie, dass die Relation *die Kurve c_i ist äquivalent zur Kurve c_k* tatsächlich eine Äquivalenzrelation ist.

3. Wie kann das Verfahren zur Schematisierung von Karten erweitert werden, sodass die Schematisierung auch Hindernisse (z.B. Punkte, Polygone) respektiert, die nicht vereinfacht werden sollen?

4. Ist das Verfahren zur Schematisierung von Karten auch anwendbar, wenn ausschließlich $\{\mathrm{HVH}, \mathrm{VHV}\}$-Pfade zur Schematisierung betrachtet werden?
5. Ist das Verfahren zur Schematisierung von Karten auch anwendbar, wenn ausschließlich $\{\mathrm{VDH}, \mathrm{HDV}\}$-Pfade zur Schematisierung betrachtet werden?

Hinweis: Ein $X_1X_2X_3$-Pfad besteht aus drei Segmenten, die jeweils vom Typ $X_i \in \{H, V, D\}$ sind. Dabei steht H für ein horizontales Segment, V für ein vertikales Segment und D für ein diagonales Segment.

Lösung:

1. Siehe Folien der Übung.
2. Damit die Relation eine Äquivalenzrelation ist, muss sie reflexiv, symmetrisch und transitiv sein. Gegeben Menge P an Punkten, zwei Kurven c_1 und c_2 sind *äquivalent*, falls es kontinuierliche Abbildung $F\colon [0,1] \times [0,1] \to \mathbb{R}^2 \setminus P$ gibt, sodass
   - $F(0,t) = c_1(t)$ und $F(1,t) = c_2(t)$ für alle $t \in [0,1]$.
   - $F(s,0) = c_1(0) = c_2(0)$ und $F(s,1) = c_1(1) = c_2(1)$ für alle $s \in [0,1]$.

   (a) **Reflexiv**: Verwende Identitätsabbildung: Für alle $s, t \in [0,1]$ setze $F(s,t) = c_1(t)$.

   (b) **Symmetrisch**: Gegeben kontinuierliche Funktion G die c_1 in c_2 transformiert. Für alle $s, t \in [0,1]$ setze $F(s,t) = G(1-s,t)$. Offensichtlich wieder kontinuierlich bezüglich $\mathbb{R}^2 \setminus P$.

   (c) **Transitiv:** Gegeben kontinuierliche Funktion F_1, die c_1 in c_2 transformiert und Funktion F_2, die c_2 in c_3 transformiert, dann transformiert $F_2 \circ F_1$ die Kurve c_1 in die Kurve c_3. Da F_1 und F_2 bezüglich $\mathbb{R}^2 \setminus P$ kontinuierlich sind, muss auch $F_2 \circ F_1$ bezüglich $R^2 \setminus P$ kontinuierlich sein.
3. Indem man die Hindernisse ebenfalls in Kurven übersetzt. Man erhält damit sowohl für die eigentlich gegebenen Kurven als auch für die Hindernisse eine Ordnung in der man sie in die Karte einfügen kann. Beim Einfügen muss natürlich sicher gestellt werden, dass die vereinfachten Kurven die Hindernisse nicht schneiden.
4. Nein, solche Pfade können nicht verwendet werden, da die Wahl nicht eindeutig ist (siehe Folien). Damit kann bei jeder Platzierung einer Kurve eine Entscheidung nötig sein, welcher der beiden Pfade tatsächlich gewählt wird. Da beide Möglichkeiten betrachtet werden müssen, entstehen damit zu viele Entscheidungsmöglichkeiten.
5. Ja, solche Pfade können verwendet werden, da die Wahl des Pfades immer eindeutig ist.