

# Protokoll Theorieprüfung

10.10.06

Fächer: Effiziente Algorithmen  
Netzwerkalgorithmen  
Automaten auf unendlichen Wörtern  
Unendliche Spiele

Dauer: ca. 35 Minuten

Note: 1.0

## Effiziente Algorithmen

Wie ist ein **Flussnetzwerk** definiert?

*Ich hab die Definition aufgeschrieben, Kapazitätsfunktion, Fluss mit Flussserhaltung und Kapazitätsbeschränkung erläutert, erzählt, dass man einen maximalen Fluss sucht.*

Da gibt es die **Ford-Fulkerson-Methode**. Dazu benötigt man das Restnetzwerk, wie ist das definiert?

*Ich hab das (etwas umständlich) erklärt*

Wie zeigt man die Korrektheit?

*Mit dem Max-Flow-Min-Cut-Theorem. Hab die drei Aussagen aufgeschrieben.*

Man zeigt das jetzt über einen Ringschluss, zwei Richtungen sind einfach, zeigen Sie mal die Richtung von b) nach c)

*Habe den Beweis genau wie im Skript gemacht, habe mich bei der Definition des Schnitts ein bisschen verhaspelt, bin aber mit einem Hinweis wieder auf die richtige Bahn gekommen.*

Das war jetzt die Korrektheit, dann ist aber noch die Laufzeit interessant. Man kann das implementieren ohne groß darüber nachzudenken..

*pseudopolynomiellen Algorithmus erwähnt, gesagt, dass er Laufzeit  $O(mC)$  hat.*

Mit Breitensuche bekommt man eine bessere Laufzeit

*Laufzeit  $O(m^2n) = O(n^5)$  hingeschrieben*

Wie zeigt man das im Detail?

*Erzählt, dass man zeigt, dass sich die Distanzen im Restnetzwerk nicht verkürzen. Erläutert, warum dass weder mit Flaschenhalkanten noch mit neu entstehenden Kanten passieren kann. Danach Schritt für Schritt erklärt, wie man zu  $n \cdot m$  Schritten kommt.*

Dann gibt es ja noch andere Varianten, z.B. den Algorithmus von Dinic. Der berechnet einen Sperrfluss im Niveaunetzwerk. Wir wollen direkt zu der effizienteren Variante kommen, der **Forward-Backward-Propagation**. Wie funktioniert die?

*Habe das Potential der Kanten und der Knoten definiert und gesagt, dass man an dem Knoten mit geringstem Potential einen Überschuss erzeugt, den man dann zur Quelle und quasi rückwärts zur Senke schiebt und dass man garantieren kann, dass der Fluss auch in der Senke ankommt, da man ja den Knoten mit kleinstem Potential genommen hat.*

Wie zeigt man jetzt die Laufzeit?

*Man legt die Kosten auf die Kanten um, und zeigt, dass man Zeit  $O(n+l)$  benötigt. Habe erstmal erklärt, wie die Fifo-Queue dazu beiträgt und dann wie man auf  $O(n+l)$  kommt. Dann sollte ich noch die Summe über die  $l_i$  bilden, um auf die Gesamtlaufzeit zu kommen. Dabei habe ich erstmal die Summe nur vor die  $l_i$  gesetzt und war ziemlich verwirrt als er mir sagte, dass da was nicht stimmen würde.*

Mir hat letzt jemand eine Frage zu dem Algorithmus gestellt und im ersten Moment war ich auch irritiert. Mal gucken, ob Sie die Frage auch beantworten können. *Er malt einen Fluss auf und zeigt auf einen Knoten.* Angenommen wir erzeugen jetzt da einen Überschuss und schieben den weiter in Richtung Senke. Es könnte ja sein, dass wir den Fluss in eine Sackgasse schieben, der Weg also überhaupt nicht zur Senke führt..

*Bin nach kurzem Überlegen schließlich darauf gekommen, dass man in dem Fall den Überschuss eben nicht an dem Knoten mit kleinstem Potential erzeugt hätte und dieser Fall somit gar nicht auftreten kann.*

Ja genau, man würde im ersten Schritt einen Überschuss der Größe 0 erzeugen und danach wäre der weg. Kommen wir nochmal zu einem anderen Thema. Online-Algorithmen - **Paging**, das hatten wir heute noch nicht. Was bedeutet  $c$ -competitive?

*Habe erklärt, dass man die Kosten des Online-Algorithmus mit denen des optimalen vergleicht und hab die Ungleichung dazu aufgeschrieben.*

Wie gut kann denn ein deterministischer Online-Algorithmus sein?

*Er kann nicht besser als  $k$ -competitive sein. (Dass man da eine möglichst schlechte Eingabe für den Online-Algorithmus konstruiert, erzählte er dann selbst.)* Irgendwie kam er dann auf Markierungsalgorithmen und ich hab erklärt was das ist und dass Markierungsalgorithmen  $k$ -competitive sind. Habe kurz den Beweis skizziert, dass ein Markierungsalg. in jeder  $k$ -Phase höchstens  $k$  und ein beliebiger Algorithmus mindestens einen Fehler pro  $k$ -Phase macht.

Wie ist das mit FIFO?

*War erstmal geschockt, weil das nur ganz kurz im Skript stand und ich erstmal in meinem Gedächtnis kramen musste. Aber es reichte dann, dass ich sagte, dass Fifo kein Markierungsalgorithmus ist, aber trotzdem competitive ist.*

Kann man denn irgendwie doch noch einen besseren Online-Alg. finden?

*Mit Randomisierung. Wir hatten da Mark, also auch ein Markierungsalgorithmus,  $2H_k$ -competitive, habe gesagt, dass er bei einem Seitenfehler eine beliebige nicht markierte Seite verdrängt.*

Das stimmt so nicht.

*Wusste überhaupt nicht, wo mein Fehler liegen sollte, hab deswegen einfach nochmal von vorne erklärt und dabei gesagt, dass Mark eine zufällige nicht-markierte Seite verdrängt.*

Ja genau, es ist entscheidend, dass er das uniform zufällig macht! Wie beweist man den competitive-Faktor?

*Man zeigt erstmal, dass opt die amortisierten Kosten  $\frac{f}{2}$  hat. Den Beweis wollte er dann noch im Detail wissen..*

Was zeigt man dann noch?

*Wie hoch die erwarteten Kosten von Mark sind. Habe erzählt, dass er schonmal  $f$  Fehler auf die frischen Seiten macht und man dann noch die Fehler auf vorherige Seiten betrachten muss. Da musste ich zum Glück nicht mehr die Rechnung zu machen, den Rest hat er wieder selbst erzählt. Dann fiel ihm wohl auf, dass es langsam mal Zeit für eine andere Vorlesung war. Ein Blick auf die Uhr sagte mir, dass ungefähr schon 20 Minuten vorbei sein mussten.*

## Netzwerkalgorithmen

Wir hatten da Wireless Overlay Netzwerke. Da hatten wir  $c$ -spanner und power spanner definiert und den Yao-Graphen betrachtet. Was ist denn erstmal ein  $c$ -spanner und ein power spanner?

*Definitionen hingeschrieben..*

Und wie sieht der Yao-Graph aus?

*Habe erklärt, dass man die Fläche in gleich große Sektoren aufteilt mit  $\alpha = \frac{2\pi}{k}$ . Habe dazu ein Beispiel mit  $k = 8$  aufgemalt und erklärt, dass die Datenübertragung immer nur zu dem nächsten Punkt im Sektor möglich ist. Habe dann auch den Routing-Algorithmus dazu erklärt. Dann kamen wir irgendwie auf die Größe von  $k$  zu sprechen und dass man nur für  $k \geq 7$  garantieren kann, dass der Algorithmus terminiert. Ich sollte dann nochmal erläutern woran man jetzt erkennt, dass der Algorithmus terminiert. Habe am Beispiel erklärt, dass der Abstand zum Ziel in jedem Schritt kleiner wird. Dann hat er noch ein Beispiel aufgemalt, wo in einem benachbarten Sektor ganz viele Knoten dicht nebeneinander waren und wollte dazu hören, dass es vom Energieverbrauch her sinnvoller wäre über die Knoten im benachbarten Sektor zu gehen, was der Algorithmus aber nicht macht.*

## Automata on Infinite Words

Wie sehen denn Büchi-Automaten im Vergleich zu endlichen Automaten aus?

*Eigentlich sehen die Automaten gleich aus, aber bei Büchi-Automaten hat man unendliche Eingabewörter und ein Wort wird akzeptiert wenn unendlich oft Endzustände besucht werden.*

Wie ist das mit der Ausdrucksstärke von nichtdet. und det. Büchi-Automaten?

*Im Gegensatz zu endlichen Automaten sind nichtdet. Büchi-Automaten mächtiger als det. Büchi-Automaten*

Können Sie eine Sprache nennen, die nicht det. Büchi-erkennbar ist?

*Deterministische Büchi-Automaten können generell nicht erkennen wenn etwas nur endlich oft vorkommen darf. (Das reichte ihm.)*

Wie sieht es mit Abschluss unter Komplement aus?

*Büchi-Automaten sind unter Komplement abgeschlossen*

Deterministische oder Nichtdeterministische?

*Obwohl ich das eigentlich wusste, war ich mir da auf einmal unsicher. Zum Glück hat er da relativ schnell das Thema gewechselt, weil er sich wohl auch nicht so sicher war.*

Wie kann man zeigen, dass Büchi-Automaten unter Schnitt abgeschlossen sind?

*Man konstruiert einen Automaten, bei dem die Zustände Tripel sind. Erste Komponente für den ersten Automaten, zweite für den zweiten Automaten und die dritte Komponente kann die Werte 1,2 oder 3 annehmen. Der Automat startet mit einer 1 in der dritten Komponente und wechselt zu 2, sobald die erste Komponente ein Endzustand ist, zu 3 sobald auch in der zweiten Komponente ein Endzustand besucht wurde usw.*

Wieso wählt man diese Konstruktion und macht das nicht genau wie bei endlichen Automaten?

*Mir war zunächst nicht klar, worauf er hinaus wollte, da mir die Konstruktion bei endlichen Automaten auch nicht mehr so geläufig war. Bin aber schließlich darauf gekommen, dass die beiden Automaten nicht gleichzeitig einen Endzustand besuchen müssen und deswegen diese Konstruktion nötig ist.*

## Unendliche Spiele

Was gibt es denn für Spiele?

*Die einfachsten Spiele sind Erreichbarkeitsspiele. Man hat eine Menge  $F \subseteq Q$  gegeben und Spieler 0 gewinnt die Partie genau dann, wenn im Laufe des Spiels ein Knoten aus  $F$  besucht wird. Habe dann die Attraktorkonstruktion erläutert.*

Wieviele Schritte braucht die Attraktorkonstruktion?

*Maximal  $|Q|$ , wenn in jedem Schritt ein Knoten hinzukommt.*

Das war es dann. Gehen Sie bitte kurz raus..

Die Prüfungsatmosphäre war angenehm, Herr Vöcking und sein Beisitzer sehr freundlich. Zu Beginn fragte Herr Vöcking, womit er anfangen solle. Dass er Effiziente Algorithmen so ausführlich geprüft hat, war wohl eher ein Zufall, passte mir aber ganz gut. Wir hatten die Vorlesung bei Herrn Rossmannith gehört, uns aber dann für das Skript von Herrn Vöcking als Grundlage entschieden, wobei wir einiges, z.B. Randomisierte Algorithmen, rausließen, da wir die praktisch nicht gemacht hatten.

Netzwerkalgorithmen war sehr mühsam zu lernen. Herr Vöcking versicherte uns vorher, dass er keine großen Rechnungen oder Abschätzungen sehen wolle. Man sollte aber in der Lage sein, die generelle Vorgehensweise bei einem Beweis zu erklären.

Bei dem Automatenzeug merkte man, dass er da gerne die Analogie bzw. Unterschiede zu endlichen Automaten suchte.

Für alle Vorlesungen hatten wir eine Themenübersicht erstellt, die er während der Prüfung auf dem Tisch liegen hatte und aus der er sich immer das nächste Prüfungsthema raussuchte.