Diplomprüfung: 1685 „Effiziente Graphenalgorithmen“

Prüfer	: Prof. Dr. Verbeek
Beisitzer	: Dr. Rettinger
Datum	: 05. Februar 2004
Dauer	: 25 Minuten
Note	: 1,7

Vorbereitung

Zur Prüfungsvorbereitung habe ich eine Zusammenfassung aller Prüfungsfragen bei Prof. Verbeek, die von Sandra Gesing stammt, überarbeitet, das Inhaltsverzeichnis als Fragenkatalog gestaltet und in TEXgesetzt. Zusammen mit dem Buch „Grapen und Algorithmen“ von Andreas Brandstädt, erschienen im Teubner Verlag Stuttgart, dessen ersten sieben Kapitel identisch mit dem Kurstext sind, habe ich für die Prüfung gelernt. Zwei Gespräche in Hagen mit Herrn Rettinger über die Schwerpunkte der Prüfung haben die Vorbereitung komplettiert.

Prüfungsverlauf

Ohne weitere Einleitung kam Prof. Verbeek gleich zur ersten Frage. Der Verlauf der Prüfung hat sich ungefähr wie folgt abgespielt:

Wie ist der greedy-Algorithmus definiert?
Da mich diese Fragen ohne weitere Aufwärmung leicht nervös erwischte, fing ich an, diesen für Graphen erklären zu wollen, worauf Prof. Verbeek sagte:

Der greedy-Algorithmus ist aber für beliebige Mengen, nicht nur für Graphen definiert.
Daraufhin habe ich Definition noch einiger Maßen hinbekommen.

Eingabe: Eine endliche Menge E und eine Kostenfunktion $c : E \to \mathbb{R}$, einen Mengensystem $J \in \mathcal{P}(E)$ mit $\emptyset \in J$.

Ausgabe: $I \subseteq J$.

Algorithmus

1. Solange noch Elemente in E sind, mache (2)
2. Entnimm ein Element $e \in E$ mit maximalen Kosten und füge es zu I hinzu, falls $(I \cup e) \in J$ ist.

Wobei sich meine Beschreibung sehr viel holpriger anhörte, da mir die exakte Definition leider nicht mehr einfiel.

Der greedy-Algorithmus liefert nur ein lokales Optimum. Wann liefert er denn ein globales Opitmum?
Wenn das Mengentupel (E, J) ein Matroid sind.

Was ist ein Matroid?
Sei E eine endliche Menge und $J \subseteq \mathcal{P}(E)$.

$M(E, J)$ heißt *Matroid* genau dann, wenn $J \neq \emptyset$ und es gelten die folgenden zwei Bedingungen:

1. Für alle $A \in J, B \subseteq A$ gilt $B \in J$. (J ist nach unten Teilmengen-abgeschlossen)
2. Für jedes $A \subseteq E$ haben alle maximalen Untermengen von A, die in J liegen, dieselbe Mächtigkeit. Diese Mächtigkeit ist *Rang* von A.

Ich hbe zwar noch die Äquivalenz der zweiten Bedingung zu dem e-Austauschelement erwähnt und begründet, habe dafür allerdings die zweite Bedingung etwas falsch wiedergegeben, was mir Prof. Verbeek nach der Prüfung erklärte. Während der Prüfung war er sich da selber nicht ganz sicher.

Sie hatten vorhin Graphen angesprochen. Nennen Sie doch mal einen konkreten greedy-Algorithmus und was ist dabei das Matroid?
Ich erwähnte die drei Verfahren von Jarnik, Prim, Dijkstra, von Kruskal und von Boruvka, Sollin.

Nehmen Sie irgend einen.
Ich habe mich für Kruskal entschieden und gesagt, dass Matroid = (Kanten, Minimalgerüste) sei.

Nur Minimalgerüste sind aber nicht nach unten Teilmengen abgeschlossen.
Ja natürlich, dann sind alle Gerüste und nicht nur die Minimalgerüste in dem Matroid enthalten.

Und wie funktioniert jetzt Kruskal?
Da habe ich den Algorithmus von Kruskal erklärt, mit der Voraussetzung, dass die Kanten nach Gewicht sortiert werden, wobei Prof. Verbeek einwarf, dass dann ja auch noch eine Kostenfunktion benötigt würde, die ich vergessen hatte. Bei der Bezeichnung der genauen Mengen, die benötigt würden, um Kruskal laufen zu lassen, hatte ich auch einige Hänger. Prof. Verbeek hat aber durch gezielte Nachfragen geholfen, diese richtig darzustellen. Dazu malte ein einen Graphen auf, der bei meiner Darstellung mit Kruskal falsch erzeugt worden wäre, worauf hin ich die richtige und dann auch genaue Definition geliefert habe.

Kommen wir nun zu unabhängigen Knotenmengen. Welche Laufzeiten haben Algorithmen zur Erkennung?
Bei beliebigen Graphen handelt es sich um das INDEPENDENT SET Problem, welches NP-vollständig ist. Paare Graphen lassen sich in Polynomzeit erkennen.

Wa ist denn NP-vollständig?
$A \in \mathrm{NP} \wedge A$ ist NP-hart.

NP-hart bedeutet, dass für alle Sprachen in NP eine polynomzeit berechenbare Funktion existiert, mittels derer diese Sprachen auf A reduziert werden können.

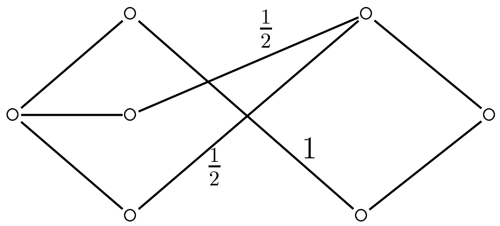
Muss das denn nicht anders herum lauten, dass A auf alle NP-Sprachen reduzierbar ist?
Da griff Dr. Rettinger ein und sagte, dass das schon richtig gewesen sei, was ich gesagt habe. Prof. Verbeek entschuldigte sich daraufhin.

Wie kann die polynomielle Laufzeit bei paaren Graphen gezeigt werden?
Mit Hilfe des DFS werden die Knoten abwechselnd gefärbt und dabei die Rückwärtskanten berücksichtigt. Das hat polynomielle Laufzeit.

Ja, das könnte stimmen. Wie sieht denn die Laufzeit bei maximal unabhängigen Kantenmengen aus?
Für beliebige und paare Graphen ist sie polynomiell. Bei beliebigen Graphen kann das mit der Ergänzungspfadmethode gezeigt werden.

Die kann man natürlich auch für paare Graphen nehmen. Aber gibt es auch andere Möglichkeiten?
Ja, mit dem Maximalflussalgorithmus von Dinitz. Dazu wird aus dem paaren Graphen ein Netzwerk konstruiert, indem die ungerichteten in gerichtete Kanten verwandelt, ein Quell- und ein Zielknoten hinzugefügt und alle Kanten mit dem Gewicht 1 versehen werden. Alle von Dinitz ermittelten Kanten mit einem Fluss von 1 sind die gesuchten.

Was ist denn bei folgendem Netzwerk? Da ist der Maximalfluss von 2 erreicht.



Das geht aus einem trivialen Grund nicht. Die algebraischen Operationen im Algorithmus von Dinitz sind lediglich Addition und Subtraktion. Bei ganzzaligen Kapazitäten können keine gebrochen rationalen Flüsse entstehen, so dass das Ergebnis bei Dinitz bzw. Ford/Fulkerson immer ganzzalig ist.

Fazit

Das Protokoll stellt natürlich keine wortgenaue Wiedergabe dar. Es gibt aber hoffentlich außer den Fragen auch in etwa die Art wieder.

Prof. Verbeek zeichent sich dadurch aus, dass er auch nicht immer alles weiß und dabei keine Probleme hat, das zu zeigen. Für den einen mag das vielleicht etwas verwirrend sein. Ich empfand das als eher beruhigend. Außerdem hat er eine für mich sehr klare Art, die Fragen so zu formulieren, dass sie auch (für mich) verständlich sind.

Wichtig ist aber, die Definitionen und Algorithmen genauestens zu kennen und zumindestens die wichtigsten Datenstrukturen richig wiedergeben zu können. Darüber hinaus ist die richtige Wortwahl sicher nicht schädlich. Mangels Übung in theoretischer Informatik habe ich häufiger Sätze wie „Teilmenge von J“ (musste natürlich „Element von J“ heißen, da J eine Potenzmenge ist) oder „Element von A“ (hier muss es natürlich „Teilmenge von A“ heißen) gesagt, was sich wohl nicht auf die Note ausgewirkt dafür umso mehr zu meiner Nervosität beigetragen hat.

Angesichts des eher mäßigen Verlaufs mit vielen Korrekturen von mir und auch von Prof. Verbeek bin ich mit der Note mehr als zufrieden. Ich kann Prof. Verbeek und diesen Kurs für eine Prüfung uneingeschränkt empfehlen.

Ich wünsche allen viel Erfolg bei den Prüfungen.

Thomas Schwarze