Prüfungsprotokoll

Effiziente Algorithmen und Theorie des Logikentwurfs

Dr. Thomas Hofmeister

11. September 2003

Note: 1.0
Prüfungsstoff: Effiziente Algorithmen 2002, Theorie des Logikentwurfs 2002

Effiziente Algorithmen

Topologisches Sortieren

- Einführend erzählt, dass eine topologische Sortierung auf einer partiellen Ordnung durchgeführt wird, welche durch einen kreisfreien, gerichteten Graph dargestellt werden kann.
- Algorithmus: Tiefensuche, bei der die Nummern von n bis 1 jeweils nach Beendigung des Aufrufs vergeben werden.
- Beweis: Wenn Kante $(v, w) \in E$, dann muss $num(v) < num(w)$ sein. Gezeigt, dass die beiden Fälle, wo dies nicht der Fall ist, nicht auftreten können, da ansonsten Widerspruch zur Kreisfreiheit bzw. zur Existenz der Kante (v, w).

Analyse der Pfadkomprimierung bei UNION-FIND-Datenstrukturen

- UNION-FIND-Datenstruktur mit schnellen UNIONs erklärt. Wurzelgerichtete Bäume aufgezeichnet, erklärt wie ein UNION funktioniert und was bei einem FIND passiert. Gesagt, dass wir die Kosten der Befehlsfolge r, bestehend aus $n-1$ UNIONs und m FINDs, mit Hilfe der Buchhaltermethode abschätzen wollen.
- Die Datenstrukturen U_r^{ohne} und U_r^{mit} eingeführt. Rang, der über U_r^{ohne} definiert ist, erklärt und Rangwachstumseigenschaft erläutert. Gesagt, dass diese auch in U_i^{mit} gilt (musste ich nicht beweisen).
- Die Zweierturm-Funktion und ihre Umkehrfunktion wollte Herr Hofmeister gar nicht mehr hören, sondern nur noch die Anwendung der Buchhaltermethode sehen. Gesagt, in welche Ranggruppe ein Element mit Rang r kommt ($\log^* r$) und dann die m FIND-Konten und n Knoten-Konten eingeführt. Erläutert, wie die $k+1$ Kosteneinheiten für ein FIND der Form $v_k \to v_{k-1} \to \cdots \to v_1 \to v_0$ verteilt werden. Grobe Abschätzung der Belastungen der einzelnen Konten gemacht. Abschließend gesagt, dass in der Laufzeit von $O((m+n)\log^* n)$ das $\log^* n$ zwar gegen ∞ konvergiert, die Laufzeit in der Praxis jedoch quasi linear ist.

Approximationsalgorithmen für das metrische TSP

- Gefragt war der Algorithmus von Christofides mit Güte $\frac{3}{2}$. Rahmenbedingungen des metrischen TSP erklärt (Kostenrelation symmetrisch und Dreiecksungleichung gilt).

- Algorithmus: Berechnung des minimalen Spannbaums S_{opt}. Gesagt, warum Kosten von S_{opt} durch Kosten der optimalen Tour π_{opt} beschränkt sind. Erläutert, dass im Gegensatz zum Algorithmus mit Güte 2 nicht alle Knotengrade verdoppelt werden. Knotenmenge M und das kostenminimale maximale Matching M_{opt} eingeführt. S_{opt} und M_{opt} werden verschmolzen. Definition eines Eulerkreises gegeben und gesagt, dass Eulerkreis EK auf verschmolzenem Graph berechnet wird. Gezeigt, wie eine Tour π aus dem Eulerkreis berechnet wird. Abschließend Beweis, dass $c(M_{opt}) \leq \frac{1}{2} \cdot c(\pi_{opt})$ gilt.

Derandomisierung mit der Methode der bedingten Wahrscheinlichkeiten

- Anhand eines abstrakten randomisierten Algorithmus A_{rand} erklärt, was das Gewicht einer Belegung der zu berechnenden Zufallsbits $y_1, \ldots, y_m$ ist. Dazu Zufallsvariable Z eingeführt, deren Wert von einer Belegung der Zufallsbits abhängt. Gesagt, dass $w(\emptyset)$ dem Erwartungswert $E(Z)$ entspricht („noch alle Möglichkeiten offen“) und dass eine komplette Belegung $w(d_1, \ldots, d_m)$ dem Wert einer konkreten Lösung entspricht. Den Algorithmus angegeben, der sukzessive die $d_1, \ldots, d_m$ berechnet. Anschließend bewiesen, dass $E(Z) \leq w(\emptyset) \leq w(d_1) \leq w(d_1, d_2) \leq \cdots \leq w(d_1, \ldots, d_m)$ gilt, da sich $w(d_1, \ldots, d_i)$ als Linearkombination von $w(d_1, \ldots, d_i, 0)$ und $w(d_1, \ldots, d_i, 1)$ schreiben lässt und damit zwischen diesen beiden Werten liegen muss.
- Nun wollte Herr Hofmeister wissen, wie sich die Gewichte bzw. bedingten Erwartungswerte bei $MAXSAT$ berechnen. Gesagt, dass der bedingte Erwartungswert linear ist und wir also die bedingten Erwartungswerte für die einzelnen Klauseln berechnen können. Drei Fälle angegeben:
  1. Alle Variablen einer Klausel schon so belegt, dass die Klausel nicht erfüllt ist. Erwartungswert für Klausel ist 0.
  2. Mindestens eine Variable so belegt, dass Klausel erfüllt wird. Erwartungswert für Klausel ist 1.
  3. Einige Variablen einer Klausel so belegt, dass diese noch nicht erfüllt ist und einige Variablen noch unbelegt. Berechne Produkt der Wahrscheinlichkeiten, dass die einzelnen unbelegten Literale nicht erfüllt werden. Eins minus dieser Wahrscheinlichkeit ist Erwartungswert für diese Klausel.

Theorie des Logikentwurfs

Baummethode

- Definition für Pinup-Polynome gegeben. Gesagt, dass das Ziel die Berechnung eines Pinup-Polynoms ist, da dieses nach Vereinfachung nur noch alle Primimplikanten enthält. UND- und ODER-Lemma sowie die beiden Fälle der Shannon-Zerlegung hingeschrieben (ohne Beweis).
- Anschließend Algorithmus hingeschrieben. Erläutert, wie p_0 und p_1 entstehen.
- Korrektheit des Algorithmus gezeigt. Polyome werden immer kürzer $\Rightarrow$ Induktionsbeweis ist günstig. Für $p = 0$ bzw. $p = 1$ werden die richtigen Pinup-Polynome berechnet. Induktionsannahme, dass für p_0 und p_1 die korrekten Pinup-Polynome p_0^* und p_1^* zurück

geliefert werden. Hier hat Herr Hofmeister dann abgebrochen, mit der Begründung, dass man ja schnell sieht, warum $p' = \overline{x}_i \star p_0^* \vee x_i \star p_1^* \vee p_0^* \star p_1^*$ dann das Richtige berechnet.

Worst case der Baummethode

- Funktion hingeschrieben, die nur 1 wird, wenn mindestens eine Zeile einer Matrix nur Einsen enthält. Erläutert, wie das Polynom für diese Funktion aussieht und warum es schon alle Primimplikanten enthält.

- Vorgehen der Baummethode skizziert und gesagt, dass diese so lange läuft, bis sie auf konstante Funktionen trifft. Egal, ob die Variablen stets auf 0 oder stets auf 1 gesetzt werden, die Funktion wird frühestens nach m Konstantsetzungen von Variablen konstant. Zerlegungsbaum skizziert und Mindestanzahl der Zerlegungen $\sum_{i=0}^{m-1} 2^i = 2^m - 1$ berechnet.

Struktursatz für OBDDs

- Erzählt, dass an einem Knoten mit Markierung x_i in einem OBDD eine spezielle Subfunktion berechnet wird, die sich durch Konstantsetzung der Variablen $x_1, \ldots, x_{i-1}$ (mit $\pi = \mathrm{id}$) ergibt. Menge S_i eingeführt, dann gezeigt, wie mit Hilfe von jeweils $|S_i|$ Knoten auf Ebene i und zwei Senken ein OBDD für die Funktion konstruiert werden kann. Den Induktionsbeweis, dass dieses OBDD tatsächlich die Funktion darstellt, musste ich nicht mehr machen.

Minimalität von OBDDs

- Herr Hofmeister sagte, er habe in seiner Vorlesung gezeigt, dass die zwei Reduktionsregeln für OBDDs ausreichen würden und ich solle das doch mal beweisen. Hab folgenden Satz bewiesen:

  Das OBDD G ist minimal. $\Leftrightarrow$ Keine der beiden Reduktionsregeln mehr anwendbar.

  Beide Richtungen wie im Sieling-Text gezeigt („$\Leftarrow$“ mit Skizze).

Achillesfersenfunktion ACH

- Definition von ACH_n hingeschrieben und gesagt, dass die Funktion quasi ein Spezialfall von DQF ist, bei der es zu jeder Variablenordnung π eine Belegung von s gibt, so dass DQF in der ungünstigen Variablenordnung berechnet wird. Dann Beweis quasi wie im Sieling-Text geführt: s-Variablen aus π streichen, mindestens $\frac{n}{2}$ der x-Variablen im linken Teil L und mindestens $\frac{n}{2}$ der y-Variablen in R. Die Mengen $J(0), \ldots, J(n-1)$ eingeführt und die Größe ihrer Vereinigung abgeschätzt ($\geq \frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4}$). Argument, dass mindestens eine Menge $J(t)$ mindestens $\frac{n}{4}$ Paare enthalten muss, gebracht. Konstantsetzung des OBDDs, so dass $|s| = t$ ist, sowie alle Variablen, die nicht in $J(t)$ vorkommen, auf 0 setzen. Es entsteht „ungünstige“ DQF über $\frac{n}{4}$ Variablenpaaren, bei der alle x-Variablen vor allen y-Variablen in π stehen $\Rightarrow$ Größe des OBDDs für Subfunktion ist $2^{\frac{n}{4}+1}$. Konstantsetzungen verkleinern OBDD höchstens, also ist ursprüngliches OBDD für ACH_n mindestens so groß.

Kommentar

Herr Hofmeister ist als Prüfer sehr zu empfehlen. Die Prüfungsatmosphäre ist ruhig und entspannt. Bis jetzt hatte ich, so weit ich mich erinnern kann, in keiner Prüfung so wenig Zwischenfragen. Herr Hofmeister lässt einen in Ruhe zum Thema antworten und bricht nur ab, wenn er anscheinend genug gehört hat. Zwischendurch gibt er zwar ab und zu das Tempo vor (siehe Zweierturm-Funktion bei der Analyse der Pfadkomprimierung), jedoch bekommt man dadurch auch schnell ein Gefühl dafür, wieviel man zu jedem Thema erzählen sollte bzw. wie tief man dabei gehen sollte.

Für diese Prüfung habe ich etwa viereinhalb Monate gelernt. Den größten Teil der Zeit dürfte dabei jedoch die Erstellung von eigenen Zusammenfassungen zu beiden Themenbereichen beansprucht haben. Auch wenn dieses Vorgehen sehr aufwändig ist, kann ich es nur weiterempfehlen. Wenn man den Stoff einmal in eigenen Worten hinschreibt, merkt man nämlich recht schnell, ob man wirklich alles verstanden hat. Desweiteren bekommt man dadurch auch schon ein Gefühl, wie man etwas in der Prüfung erklären kann. Bei der Wiederholung ist die Verwendung von eigenen Zusammenfassungen ebenfalls sehr praktisch, da in diesen alles so formuliert ist, wie man es selbst am besten versteht, so dass man über viele Dinge, die beim ersten Mal relativ schwer zu verstehen sind, beim zweiten Lesen nicht mehr lange nachdenken muss.

Die Antworten zu den einzelnen Themen habe ich nicht so ausführlich geschrieben, da ich nur grob skizzieren wollte, was ich zu den einzelnen Themen erzählt hab. Genauere Antworten finden sich unter anderem in den etwas weiter oben erwähnten Zusammenfassungen, welche unter der folgenden Adresse zu finden sind:

`http://www.michaelgregorius.de/`

Viel Glück für die Prüfung!