

PRÜFUNGSprotokoll "THEORETISCHE INFORMATIK"

Matthias Sondermann

6. Oktober 2004

Effiziente Algorithmen	(Buch, 2nd Edition)	geprüft von Prof. Hromkovic
Termeresetzungssysteme	(Vorlesung 2004 von Prof. Giesl)	geprüft von Prof. Giesl
Logikprogrammierung	(Vorlesung 2003 von Prof. Indermark)	geprüft von Prof. Giesl

Dauer: 35 Minuten

Note: 1,0

Die Prüfung ist nun schon ein paar Tage her und ich merke gerade, dass ich mich schon kaum an den genauen Inhalt erinnern kann. Deswegen nehmt das hier alles nicht wortwörtlich. Ich hoffe, ich kann dem einen oder anderem mit diesem Protokoll ein wenig helfen. Mir persönlich haben die Gedächtnisprotokolle in der Prüfungsvorbereitung sehr genutzt. Jetzt aber zum Inhalt:

Hromkovic: Wer sucht denn die Reihenfolge aus?

Giesl: Der Herr Sondermann darf selbst entscheiden.

Ich: Ich fang mit Effiziente an.

Effiziente Algorithmen (10 Minuten)

Hromkovic: Erzählen Sie doch mal was zu Divide and Conquer und zum Dynamischen Programmieren.

Ich: Grobe Definition der beiden Design-Konzepte angegeben, dann Unterschiede (top-down, bottom-up) und Gemeinsamkeiten (beide setzen Teillösungen zur Gesamtlösung zusammen) erwähnt. Das mit dem Zusammensetzen schien ihm wichtig zu sein. Da ich zu beiden Techniken direkt Beispiele genannt hatte (weil die Frage in jedem Protokoll steht), reichte ihm das hierzu auch schon.

Hromkovic: Sie erwähnten gerade einen Rucksackalgorithmus.

Ich: Ja, das ist ein pseudopolynomieller Algorithmus, der auf Tripeln (k, W, T) aufbaut. Hab dann kurz erklärt, was die Tripel bedeuten und wie der Algorithmus funktioniert. Wichtig dabei: Wo steckt darin das dynamische Programmieren? In der Berechnung von TRIPLE_{i+1} aus TRIPLE_i .

Hromkovic: Und wieso ist das jetzt ein pseudopolynomieller Algorithmus?

Ich: Erstmal definiert, was ein pseudopolynomieller Algorithmus ist (Integer-Valued-Problem, MaxInt, Komplexität). Dann nur erwähnt, dass $|\text{TRIPLE}| \leq n \cdot \text{MaxInt}(x)$. Das reichte dann auch schon wieder.

Hromkovic: Wann gibt es denn keinen pseudopolynomiellen Algorithmus?

Ich: Wenn das Problem stark NP-schwer ist, also ein Polynom p existiert, so dass $\text{Value}(p)$ NP-schwer ist. Ups, hab ja noch gar nicht gesagt, was ein $\text{Value}(p)$ -Problem ist, also mal schnell nachholen.

Hromkovic: Kennen Sie ein stark NP-schweres Problem?

Ich: Ja, SAT. Ups vertan, das Beispiel im Buch war ja mit TSP. Direkt verbessert, Reduktion $\text{HC}_{\leq p} \text{Lang}_{\text{Value}(p)}\text{-TSP}$ erwähnt und Konstruktion erklärt.

- Hromkovic:** Themawechsel: Was ist denn ein Polynomialzeit Approximationsschema?
- Ich:** Hä? Ein was? Erstmal 5 Sekunden Stille im Raum, bis ich bemerkte, dass ich immer nur mit dem Begriff PTAS gelernt hatte und eigentlich schon wußte was das ist. Also Definition heruntergeleiert.
- Hromkovic:** Dazu hatten wir doch ein Beispiel, oder?
- Ich:** Ja, ein PTAS für das einfache Rucksackproblem. Ohne Umschweife direkt Algorithmus erklärt und mit Komplexität angefangen, aber davon wollte er nichts wissen.
- Hromkovic:** Und wie kann man zeigen, dass der Fehler nicht größer als ε ist?
- Ich:** Güteabschätzung aus dem Buch angegeben (also M, P, P*). Irgendwann hat er dann abgebrochen.
- Hromkovic:** Ok, der Rest ist ja nur noch Rechnung. Was ist denn ein duales Problem?
- Ich:** Schluck! Öhm, das Kapitel hat soweit ich weiß noch nie jemand in seinem Protokoll erwähnt, weshalb ich es auch beim Lernen "etwas" vernachlässigt hab. Also hab ich angefangen, irgendwas über Lineares Programmieren zu erzählen und erwähnt, dass man mit dem Dualproblem, eine obere bzw. untere Schranke für die eigentliche Lösung berechnen kann (dies war der erste und einzige Moment, wo mein Nebenfach BWL mit Operations Research Sinn gemacht hat).
- Hromkovic:** Ok, ich sehe, Sie können das. Haben Sie sich auch mit randomisierten Verfahren beschäftigt?
- Ich:** Puh, nochmal Glück gehabt.
Ja, es gibt da einen randomisierten Multistart-Lokale-Suche-Monte-Carlo-Algorithmus für das 3SAT-Problem. Lokale Transformation angegeben. Anscheinend ist ihm hier wirklich nur die Idee des Algorithmus wichtig. Ihn hat dann nur noch p_j und q_{ji} interessiert.
- Hromkovic:** Ok, das wars von meiner Seite.

Termersetzungssysteme (15 Minuten)

- Giesl:** Schreiben Sie doch mal ein Gleichungssystem auf, dass "plus" berechnet!
- Ich:** Darf ich für *plus* und *succ* die kürzeren Schreibweisen p und s benutzen?
- Giesl:** Ja klar, ist ja viel übersichtlicher.
- Ich:** Ok, also $p(0, y) \equiv y$ und $p(s(x), y) \equiv s(p(x, y))$.
- Giesl:** Was will bzw. kann man jetzt damit machen?
- Ich:** Man kann überprüfen, ob andere Gleichungen daraus folgen.
(Folgerungsbegriff erklärt und Beweisrelation erwähnt)
- Giesl:** Folgt denn die Gleichung $p(x, 0) \equiv x$?
- Ich:** Erstmal baff. Die Frage kam unerwartet und ich sah auf einmal den Wald vor lauter Bäumen nicht. Hab dann im Kopf irgendwie versucht, die beiden Seiten zu reduzieren. Nach einiger Zeit: Hmmm, nee ich glaub die folgt nicht.
- Giesl:** Ok, aber soll ja im Moment auch nur eine Vermutung sein. Es gibt ja bestimmte Methoden, um das leichter einzusehen.
- Ich:** Ja, das Wortproblem ist zwar im allgemeinen unentscheidbar, aber wenn man ein äquivalentes und konvergentes Termersetzungssystem hat, reicht es die beiden Terme der Gleichung auf ihre Normalformen zu reduzieren und auf Gleichheit zu überprüfen.

- Giesl:** (Nimmt sich mein Blatt und schreibt)
 $s \equiv_{\mathcal{E}} t \Leftrightarrow s \downarrow_{\mathcal{R}} = t \downarrow_{\mathcal{R}}$, wenn \mathcal{R} konvergent und zu \mathcal{E} äquivalent ist. Gilt das?
- Ich:** Ja klar. Der Beweis von rechts nach links ist ja einfach. Und die Richtung von links nach rechts hat bestimmt was mit der Beweisrelation zu tun, aber ...
- Giesl:** Ja alles richtig, aber was ist hier die wichtige Eigenschaft?
- Ich:** Ahhh, die Church-Rosser-Eigenschaft, die ja durch die Konfluenz gegeben ist.
- Giesl:** Ja genau. Kommen wir zu unseren Plus-Gleichungen zurück. Was kann man da jetzt zur Terminierung sagen?
- Ich:** Die Terminierung ist im allgemeinen unentscheidbar. Aber wenn ich eine Reduktionsrelation finde, so dass ich damit alle Gleichungen richten kann, bin ich fertig. Hier könnte man die lexikographische Pfadordnung ...
- Giesl:** Das stimmt schon, aber wir ändern das noch ein bisschen ab. Vertauschen Sie bei der zweiten Gleichung auf der rechten Seite doch mal das x und y. Dann wirds wenigstens etwas interessanter.
- Ich:** Ok, dann kann ich mit lpo hier nichts mehr tun. Aber die rekursive Pfadordnung ist hier geeignet. (Gleichungen gerichtet und kurz die Multimengenrelation erwähnt, dann kam der große Schock)
- Giesl:** Ahh Multimengenrelation, ist die fundiert, wenn die normale ... ach nee, das lassen wir an dieser Stelle mal.
- Ich:** (Jetzt ist alles vorbei. Er traut mir noch nicht mal den Beweis zu, der bis jetzt in fast jedem Protokoll stand. Das kann ja heiter werden. Im nachhinein wars wohl doch nicht so schlimm.)
- Giesl:** Was muss den jetzt noch gezeigt werden?
- Ich:** Die Konfluenz.
- Giesl:** Ist die entscheidbar? Warum ist die Terminierung hier wichtig?
- Ich:** Die Terminierung ist wichtig, da beim Algorithmus CONFLUENCE die kritischen Paare auf ihre Normalformen reduziert werden. Würde das TES nicht terminieren, könnte eine Reduktion unendlich sein und der Algorithmus die Konfluenz nicht mehr entscheiden. Da wir die Terminierung aber schon gezeigt haben, reicht es nach dem Diamondlemma die lokale Konfluenz zu betrachten, also nur kritische Situation (Definition einer kritischen Situation hingeschrieben). Da hier aber kein kritisches Paar existiert, ist das Termersetzungssystem konfluent und damit konvergent.
- Giesl:** Also, folgt jetzt die Gleichung von oben aus dem Gleichungssystem?
- Ich:** Nein. Die beiden Normalformen sind unterschiedlich.
- Giesl:** Aber das ist doch komisch. Das müßte doch eigentlich gelten? Wir reden doch hier von "plus"!?
- Ich:** Ja, aber das Problem ist, dass wir "plus" sofort mit den natürlichen Zahlen verbinden. Und hier gilt ja erstmal ganz allgemein die Folgerungsbeziehung. D.h. also es existieren Modelle, in denen das halt so nicht gilt.
- Giesl:** Und was kann man da jetzt machen?
- Ich:** Dafür gibt es den Begriff der induktiven Gültigkeit. (Definition aufgeschrieben, Konvergenzbeweismethode erklärt, Definitionsprinzip erwähnt)
- Giesl:** Wie kann man das algorithmisch lösen?
- Ich:** Mit dem verbesserten Vervollständigungsverfahren, das ein wenig verändert bzw. erweitert werden muss. Die "orientieren" Regel reicht so nicht mehr ganz aus. Ausserdem werden zwei neue Regeln "Inkonsistenz" und "Injektivität" eingeführt.
- Giesl:** Dann zeigen Sie doch mal, dass die Gleichung von oben wenigstens induktiv gültig ist!

Ich: (Algorithmus angewendet, also die beiden kritischen Paare gebildet, reduziert und gelöscht) Ok, da der Algorithmus nicht "false" ausgegeben hat, folgt also die Gleichung induktiv.

Giesl: Ja, das haben sie schön gemacht (;-)). Kommen wir zur Logikprogrammierung.

Logikprogrammierung

Giesl: Ok, schreiben Sie doch mal das eben erwähnte "plus" als Logikprogramm!

Ich: $p(0, X, X)$.
 $p(s(X), Y, s(Z)) : \neg p(X, Y, Z)$.

Giesl: Jetzt gibt es aus Effizienzgründen in Prolog natürlich auch Möglichkeiten, dass besser zu machen.

Ich: (Diese Frage hatte ich nun wirklich nicht erwartet, und musste mir das erstmal selbst versuchen zusammenzubasteln. Was ich da nun wirklich hingeschrieben habe, weiss ich nicht mehr. Aber als ich mit dem "is" Prädikat anfing, reichte ihm das schon. Er wollte dann noch wissen, was dieses Prädikat nun genau macht, d.h. also rechte Seite auswerten und dann mit linker unifizieren, im Gegensatz zu "=" oder "=:=".)

Giesl: Was können denn in Prolog für Indeterminismen auftreten?

Ich: Bei der SLD-Resolution kann es möglich sein, mit verschiedenen Programmklauseln zu resolvieren (ND 1.Art) und mit verschiedenen Literalen (ND 2.Art). Dabei ist der Nichtdeterminismus 2. Art ein "don't care"-Nichtdeterminismus, weil das Vertauschungslemma sagt, dass die Reihenfolge der Literale, mit denen resoliert wird, unerheblich für die Lösung ist. Prolog wendet daher die kanonische SLD-Resolution an. Der ND 1.Art kann so einfach nicht aufgelöst werden. Prolog behilft sich hierbei der Strategie, die Programmklauseln in einer bestimmten Reihenfolge zu betrachten.

Giesl: Wieso kann denn auch der Nichtdeterminismus 2.Art kritisch sein?

Ich: (erstmal Stille im Raum) Ah, es gibt bestimmt Beispiele, bei denen "fail" ausgegeben wird oder die Resolution nicht terminiert, je nachdem in welcher Reihenfolge die Literale benutzt werden??? Stimmt das??? (Da war ich mir überhaupt nicht sicher)

Giesl: Ja genau. Zum Beispiel beim Programm $q : \neg q$ und der Anfrage p, q .

Ich: Hmm stimmt. Hätte man drauf kommen können.

Giesl: Was für Möglichkeiten gibt es denn in Prolog das Backtracking zu beeinflussen? (Die Frage war irgendwie anders und komisch gestellt, so dass ich erst nicht wusste, worauf sie zielte)

Ich: Da gibt es den Cut. Er bietet dem Programmierer die Möglichkeit, Backtracking zu verhindern, so dass ganze SLD-Teilbäume nicht besucht werden.

Giesl: Wo könnte man das zum Beispiel einsetzen?

Ich: Am Beispiel einer If-Bedingung könnte ich das gut erklären. (Wollte gerade anfangen, aber da hat es schon gereicht)

Giesl: Nochmal zurück zur Theorie. Was können Sie denn zur Semantik eines Logikprogramms sagen?

Ich: Es gibt die prozedurale, die deklarative und die Fixpunktsemantik. (Alle Definition aufgesagt und kurz erklärt wofür sie gut sind)

Giesl: Sie erwähnten bei der Fixpunktsemantik *trans*. Was ist das denn genau und wofür gibts das?

Ich: *trans* ist definiert als Menge ... (Definition gegeben und sehr "informell" erklärt was das ist, also die Menge der logischen Implikationen aus der Menge M bezüglich des Logikprogramms)

Giesl: Ich denke, dass sollte reichen. Warten Sie bitte kurz draussen.

Fazit

Ich kann nur allen Recht geben, die in ihren Protokollen von einer "angenehmen Atmosphäre" sprechen. Es ist wirklich so! Der Prüfer hilft einem wo er nur kann und verzeiht kleinere Fehler ohne darauf herumzureiten. Sowohl Prof. Giesl als auch Prof. Hromkovic hetzen einen nicht und lassen auch immer ein wenig Bedenkzeit zu, wenn es mal hakt. Also: Keine Sorgen machen. Wenn man gut vorbereitet ist, kann eigentlich nicht viel schief gehen. Und noch ein kleiner Tipp: Man sollte dringend das Formulieren üben. Das hat mir unheimlich viel gebracht. Einfach zu zweit hinsetzen und sich gegenseitig befragen. Denn wie Prof. Hromkovic mir einen Tag vor der Prüfung sagte: "Wissen reicht nicht, man muss es auch präsentieren können". Insgesamt kann ich die beiden Prüfer nur empfehlen. In diesem Sinne, viel Erfolg bei Euren Prüfungen.