Diplomprüfung

Gedächtnisprotokoll
Theoretische Informatik

Prüfer: Prof. Juraj Hromkovic
Fächer: Effiziente Algorithmen,
Approximative und Randomisierte Algorithmen
Angewandte Automatentheorie
Datum: 08.04.2005
Name: Sven Ritzerfeld
Note: 1.0

Hro Beginnen wir mit parametrisierter Komplexität. Was fällt Ihnen dazu ein? (Mist, das hab ich mir nur *sehr* kurz angeguckt...)

Ich Hmja, also, da wählt man einen Parameter k, so dass man die Laufzeit mit einer Funktion, die von k abhängt, und einem p(|x|) verbessern kann.

Hro Hm, wie denn genau?

Ich ...

Hro Stellen sie doch mal die Verbindung zu den Pseudopolynomiellen Algorithmen her.

Ich Öhm, also da ist die Laufzeit polynomiell in der Eingabelänge und dem MaxInt beschränkt. Dann gibt es das Value(h) Problem, wo man dann den MaxInt durch ein Polynom über die Eingabelänge beschränkt.

Hro Dann geben die doch mal ein Beispiel zu den Parametrisierten.

Ich Da gab es zwei Stück zum VertexCover. Eines war rekursiv. Bei dem ersten hat man nach einem Parameter k parametrisiert, welcher der Größe des VertexCovers entspricht. Nun nimmt man alle Knoten mit Grad größer k ins VertexCover, weil diese enthalten sein müssen. Dann sucht man ein entsprechend kleineres VertexCover im Restgraphen.

Hro Und wo ist der Vorteil dabei? ich meine, der Subgraph kann ja immer noch sehr viele Knoten enthalten.

Ich Hmja, das stimmt...

Hro Was war denn die zweite Beobachtung?

Ich Hm, also...

Hro Wenn ich ein VertexCover der Größe maximal m habe, und die Knoten haben dann noch maximal Grad k, wie große kann mein Graph denn dann überhaupt sein?

Ich Naja, also... m mal k würd ich sagen...

Hro Ja, fast :) m mal k+1... und das zweite Beispiel?

Ich Das war nach Divide&Conquer, also rekursiv.

Hro Und wie geht man da vor?

Ich Hm...man nimmt sich einen Knoten raus, entfernt den und alle inzidenten Kanten, und macht dann in den entstehenden Subgraphen weiter.

(leider nicht so ganz richtig ;)

Hro ...was entfernt man denn da genau...und worin macht man weiter?

Ich Tja, also...ist nicht gerade mein Spezialgebiet :)

Hro Ja...also, man nimmt sich eine Kante, entfernt diese und die dazugehörenden Knoten, und macht dann weiter.

Ich Achso.

Hro Machen wir weiter...wir hatten da ein Thema, in dem es darum ging, die Laufzeit etwas weiter zu verbessern...

Ich Sie meinen Lowering WorstCase Complexity?

Hro (nickt)

Ich Ja, da hatten wir das Beispiel, wo man 3SAT in $27r*(1{,}84^n -1)$ mit Divide&Conquer lösen kann.

Hro Können sie das näher erläutern?

Ich Ja, man bekommt da die Rekurrenzgleichung T(n,r)=...

Hro (unterbricht) Mir geht es eher um den Algorithmus.

Ich Achso, ja, man führt eine neue Formel F(l=1) und F(l=0) ein, welche der Belegung der jeweiligen Variablen entspricht. Tritt l in einer Klausel nicht-negiert auf, so wird die komplette Klausel entfernt. Tritt l negiert auf, wird nur das Literal entfernt.

Hro Und wenn eine Klausel nur aus dem negierten Literal l besteht?

Ich Dann ist die komplette Formel unerfüllbar.

Hro Gut.

Ich Der rekursive Algorithmus bricht ab bzw. berechnet deterministisch alle Möglichkeiten, falls die Formel nur noch aus 3 Literalen oder 2 Klauseln besteht. Ansonsten gibt es 3 Möglichkeiten, je nach der Anzahl der Literale in den Klauseln. Für en Literal wird F(l=1) berechnet, für zwei Literale $F(l_1=0, l_2=1)$ und für drei Literale $F(l_1=0, l_2=0, l_3=1)$
(die Rekurrenzgleichung wollte er bei mir nicht sehen).

Hro Gut, wir hatten später bei den Randomisierten noch einen Algorithmus, der noch etwas besser ist, Schönings Algorithmus
(Hui, davon konnte ich auch nur den Anfang wirklich gut...)

Ich Ja, der wiederholt maximal $O(\text{Wurzel}(n)*(4/3)^n)$ mal die folgende Prozedur: Eine zufällige Lösung wird generiert, und dann wird in maximal 3n Schritten der Lokalen Suche weitergesucht.

Hro Können sie das genauer erklären?
(eher nicht... ;)

Ich (hab den kleinen Graphen aus dem deutschen Buch aufgemalt)
Ja, man hat die Wahrscheinlichkeit von >1/3 dass man sich auf die optimale Lösung α^* hinzubewegt, und <2/3 dass an sich von Ihr entfernt. Macht man also i Schritte in die falsche Richtung, muss man auch wieder i Schritte in die Richtige machen, und dann noch j bis α^*. Dann kommt man auf j+2*i über i Möglichkeiten.

Hro Und weiter?

Ich Ja...dann multipliziert man noch $(1/3)^{j+i}$ und $(2/3)^{i}$ dran, dann hat man die Wahrscheinlichkeit.

Hro Gut, das reicht mir.
(puh! :)

Hro Können die ein Approximations-Schema definieren?

Ich Ja, da gibt es einmal das PTAS und das FPTAS. Bei beiden ist die Laufzeit polynomiell über die Eingabelänge, und beim FPTAS auch über den Fehler ε^{-1}

Hro Wir hatten da ein Beispiel. Das Rucksackproblem.

Ich Ja, genau, den Knapsack.
(Hab dann den Algorithmus und Beweis der δ-Approx. kurz hingeschrieben, dann gings an den PTAS, kurz den Algorithmus erklärt)
Dann hat man da die optimale Menge M und P. P* ist die durch den Algorithmus erweiterte. Der Rest ist ähnlich wie bei dem Beweis für die δ-Approx. Das Element w_{iq} passt nicht mehr in P*, somit kann man die kleine Anschätzung machen wie oben, nur mit (k+1) im Nenner.
(Hab das noch kurz hingeschrieben)

Hro Danke, das reicht mir schon
(wollte die cost(M)/cost(S*) Rechnung nicht sehen)
Können sie mir ein Bespiel für eine Reduktion bei den Approximierten geben?

Ich Ja, beispielsweise die APX-Reduktion für Optimierungsprobleme. Da wird mit Mapping-Funktionen F und H die Eingabe bzw. das Ergebnis für A2 umgewandelt. A2 muss dabei mit einem Fehler von nur $\varepsilon=\delta/\alpha$ arbeiten, da sich die Approximation durch F und H ver-α-facht. So arbeitet A1 dann mit einem Fehler von δ.

Hro Ein Beispiel?

Ich Da wäre MaxSat auf MaxClique. Das läuft eigentlich ähnlich ab, wie bei der normalen polynomialzeit-Reduktion. Aus der Formel wird ein Graph gebastelt, so dass die Knoten Tupel (c,l) sind, also aus Klausel und Literal bestehen. Dann werden alle Knoten verbunden, die aus unterschiedlichen Klauseln sind, und nicht die Negation des Literals darstellen. Umgekehrt wird dann aus der maximalen Clique eine Belegung der Formel gemacht.

Hro Danke, das genügt. Noch ein Beispiel für eine Reduktion?

Ich Hm, was war da noch für ein Beispiel...

Hro Nein, ich meine für eine andere Reduktion. Die GAP-Reduktion.

Ich Achso, ja. Dabei wird für ein Optimierungsproblem ein GAP-c,s-Entscheidungsproblem erstellt. Dabei soll $c < Opt\_U(x)/|x| < s$ gelten. Von dem Ergebnis kann man dann auf die Existenz eines c/s-approx. Alg. schließen. Bei der GAP-Reduktion wird die GAP 'preserved', und so kann man einen Vergleich der beiden Probleme erstellen.

Hro Danke. Ein Beispiel?
(Oh mann...)

Ich Öhm, ja...das war... MaxEkLinMod auf MaxkSat?...

Hro (zuckt mit den Schultern und grinst :)

Ich Ja, doch, das war das. Da erstellt man zu jeder Gleichung $x_1+x_2+x_3=a(\mathrm{mod}\ 2)$ eine Formel Φ_a.

Hro Wie sehen die aus?

Ich Also, die Formeln sollen für jede entsprechende Belegung mit 1 wahr werden, falls die Gleichung erfüllt ist. Φ_0 ist also ($/x_1$ OR x_2 OR x_3) AND (x_1 OR $/x_2$ OR x_3) AND (x_1 OR x_2 OR $/x_3$) AND ($/x_1$ OR $/x_2$ OR $/x_3$) und dann das gleiche für Φ_1.

Hro Danke. Wir hatten bei den Randomisierten zwei Algorithmen für MaxSat...

Ich ...hm...MaxSat.....achso, das mit der Linearen Programmierung?

Hro Ja...zufälliges Runden...

Ich Ja, genau. Dabei teilt man die Variablen zu erst in die Indexmengen F^+ und F^- ein, jeweils für normale und negiert vorkommende Literale. Dann hat man die Gleichung Sum[i aus F^+] y_i + Sum[i aus F^-] y_i >= z_i, so dass z_i die Erfüllung der Klausel angibt. Das ganze wird dann in ein RelLP Problem relaxiert, so dass nicht nur Integerwerte sondern reelle Werte zulässige Lösungen sind. Das zufällige Runden des Algorithmus funktioniert so, dass eine Zufallszahl zwischen [0,1] generiert wird. Wenn diese Zahl im Intervall [0,$\alpha(y_i)$] liegt, so wird y_i auf 1 gesetzt, und auf 0 sonst. Somit entspricht die Zufallszahl der W'keit die Variable mit 1 zu belegen.

Hro Ja, dann hätte ich noch gerne die Wahrscheinlichkeit für die Erfüllung einer Klausel.

Ich Das ist (1-PI(1-$\alpha(y_i)$)). $\alpha(y_i)$ ist die W'keit, dass ein Literal mit 1 belegt wird, also ist (1-$\alpha(y_i)$) die W'keit, dass ein Literal mit 0 belegt wird. Die Multiplikation über alle Literale einer Klausel ergibt dann die W'keit, dass alle Literale gleich 0 sind. Die Gegenwahrscheinlichkeit davon ist somit die W'keit, dass eine Klausel erfüllt ist, also mindestens ein Literal gleich 1 ist. Von der Formel kommt man dann auf $(1-(1-k)^k)$).

Hro Danke, das war gut. Kommen wir zu der Automatentheorie. Was fällt Ihnen denn zur Bisimulation ein?

Ich Die Bisimulation ist feiner als die Sprachäquivalenz und wird zur Minimierung von NEAs verwendet. Dies geschieht im so genannten Markierungsalgorithmus. Dort werden zuerst die Zielzustände, welche keine ausgehenden Kanten haben, markiert. Dann geht man einen Schritt weiter, und betrachtet Knotenpaare, von denen aus man diese Zielzustände erreichen kann. Dann die Knoten, von denen aus man diese erreichen kann, und so weiter. Falls im letzten Schritt kein Knotenpaar markiert wurde, terminiert der Algorithmus.
(hab das direkt dazu gesagt, weil ich mal gehört hab, er würde gerne nach der Terminierung der Algorithmen fragen)

Hro Gut. Sie hatten in der Vorlesung ein Erreichbarkeitsproblem...

Ich ...(überleg)...achja! Das pre*(C) und post*(C) bei den Kellerautomaten. Das erkennt man durch den Saturierungsalgorithmus 1 und 2. Der erste funktioniert grob gesehen so, dass man wenn es eine Transition von p nach q gibt, und eine kante von q nach q' im Automaten existiert, eine Kante von p nach q' einfügen darf. Der zweite funktioniert ähnlich, nur dass hier eine Transition von p nach q und eine Kante von p nach q' existiert, und man so im Automaten eine Kante von q nach q' einfügen darf.
(hab das kurz so aufgezeichnet)

Hro Danke, warten sie bitte einen Moment draußen

...

Hro Ja...sie haben und etwas Kopfschmerzen bereitet... Den Prüflingen, die sich für die Randomisierten entscheiden, räumen wir immer einen Joker ein, weil dieses Thema sehr anspruchsvoll ist. Sie haben sich diesen Joker schon zu Beginn bei einem sehr einfachen Thema genommen, nämlich bei der Parametrisierten Komplexität. Dafür konnten sie aber später dann wirklich alle schweren Sachen. Daher ist es immer noch eine glatte 1.0

Generelles zur Prüfung:
Wie auch die meisten anderen, muss auch ich sagen, dass die Prüfung bei Prof. Hromkovic wirklich sehr unstressig war. Er strahlt eine angenehme Ruhe aus, so dass man nicht wirklich nervös wird. Ich musste eigentlich keinen Beweis komplett bis zum Ende aufschreiben, es hieß immer vorher "Danke, das genügt mir" (was mich doch bei dem ein oder anderen Beweis gerettet hat ;)
Der Freund, mit dem ich zusammen gelernt habe, war direkt vor mir dran, und Herr Hromkovic wusste, dass wir zusammen gelernt haben. Ich weiß nicht ob es deswegen war, aber er hat mich jedenfalls komplett andere Sachen gefragt, es war wirklich keine Frage gleich.
Denen, sie sich auch für die Randomisierten entscheiden, kann ich nur raten, sich das deutsche Buch mal auszuleihen. Hier sind viele Sachen doch ein gutes Stück ausführlicher erklärt, was manchmal schon was ausmachen kann.

Allen zukünftigen Prüflingen viel Erfolg, es ist wirklich halb so wild, nur vorher nicht zuuu verrückt machen :)
(ich weiß, ich hab gut reden, ich hab’s ja hinter mir ;P)