Gedächtnisprotokoll zur Diplomprüfung THEORETISCHE INFORMATIK

Fächer: Effiziente Algorithmen (Hromkovič)
Approximative & Randomisierte Algorithmen (Hromkovič)
Angewandte Automatentheorie (Thomas)
Prüfer: Juraj Hromkovič
Datum: 16.10.2003
Note: 1.0

Juri: *Bei der NP-Vollständigkeit gibt es diese Master-Reduktion, um zu zeigen, dass* SAT *NP-schwer ist. Was ist die Ein- bzw. Ausgabe dieser Reduktion?*

ich: Eingabe ist eine Turingmaschine und ein Wort $w \in \Sigma^*$, Ausgabe ist eine Formel φ, die genau dann erfüllbar ist, wenn $w \in L$ für ein $L \in \mathrm{NP}$.

Juri: *Können Sie die Reduktion?*

ich: Ja. (Reduktion gemacht, bis er mich unterbrochen hat)

Juri: *Erklären Sie den Floyd-Algorithmus als Beispiel zur dynamischen Programmierung.*

ich: Floyd-Algorithmus dient zur Bestimmung der kürzesten Pfade zwischen je zwei Punkten in einem Graphen. Man berechnet für $k = 0, \ldots, n$ und alle Knotenpaare (i, j) den Wert $\mathrm{cost}_k(i, j)$, der den minimalen Kosten des Pfades zwischen i und j entspricht, wobei die internen Knoten des Pfades in $\{v_1, \ldots, v_k\}$ sind.

Juri: *Wir hatten auch „parametrisierte Komplexität", was ist das?*

ich: Das ist ein Design-Konzept, bei dem man exponentielle Worst-Case Laufzeit in Kauf nimmt, wenn die Laufzeit für die in der Anwendung auftretenden Instanzen des Problems noch akzeptabel sind. Eine Parametrierung ist eine Abbildung...

Juri: *Beipiele?*

ich: VC (Divide & Conquer Strategie erläutert)

Juri: *Für* VC *hatten wir auch noch einen anderen Ansatz, wie lief der?*

ich: (die beiden Lemmata von dem anderen Ansatz erläutert)

Juri: *Kommen wir zur lokalen Suche. Was ist das?*

ich: Lokale Suche beruht auf Nachbarschaften, eine Nachbarschaft ist eine Abbildung f_x: $\mathcal{M}(x) \to Pot(\mathcal{M}(x))$, die reflexiv und symmetrisch ist und es von jeder Lösung in $\mathcal{M}(x)$ einen Pfad zu jeder anderen Lösung über Nachbarschaften gibt. Definition einer Nachbarschaft üblicherweise durch Angabe einer „lokalen Transformation", z. B. *FLIP* für SAT oder *k-EXCHANGE* für TSP. Lokale Suche bleibt in lokalen Optima hängen.

Juri: *Können Sie mir positive bzw. negative Beispiele sagen?*

ich: Optimal bei MST, sehr schlecht bei TSP.

Juri: *Für* TSP *gibt es ein „pathologisches Beispiel", dass lokale Suche schlecht ist. Was hat dieses Beispiel für Eigenschaften?*

ich: Das Beispiel beruht auf diesen Diamanten; man verbindet k Stück, so dass es genau eine optimale Lösung gibt, exponentiell viele zweitbeste Lösungen, die exponentiell schlechter sind, und jede zweitbeste Lösung ist ein lokales Optimum bzgl. einer $(3k-1)$-*EXCHANGE* Nachbarschaft.

Juri: *Können Sie das Beispiel konstruieren?*

ich: Ja.

Juri: *Es gibt auch Kernighan-Lin Variable-Depth Search, das lokale Optima wieder verlässt. Was ist das?*

ich: Lokale Suche kombiniert mit Greedy-Strategie (erklärt)

Juri: *Wir haben auch eine randomisierte Variante, um lokale Optima wieder zu verlassen, kennengelernt.*

ich: ?

Juri: *Simulated Annealing...*

ich: Ach so. (diese Analogie zur Festkörperphysik erläutert)

Juri: *Können Sie mir die Wahrscheinlichkeit sagen, mit der ich eine Verschlechterung in Kauf nehme?*

ich: Nein, irgendeine e-Funktion, wo irgendwo die Boltzmann-Konstante auftritt.

Juri: *Die Boltzmann-Konstante interessiert hier am wenigsten, wie hängt die Wahrscheinlichkeit mit der „Temperatur" zusammen?*

ich: hohe Temperatur $\rightarrow$ hohe Wahrscheinlichkeit, einen schlechteren Zustand anzunehmen; außerdem werden die Sprünge in die falsche Richtung mit sinkender Temperatur kleiner. (für das Zweite hat er extra nochmal nachgehakt)

Juri: *Bezüglich der Güte von lokaler Suche gibt es ja so eine Sache, wo sie gut läuft...*

ich: Wenn wir eine exakte, polynomzeit-durchsuchbare Nachbarschaft haben. (beides erklärt)

Juri: *Dann kommen wir mal zum randomisierten Teil. Wir hatten zwei Algorithmen für* SAT.

ich: (Scheiße)

Juri: *Welche sind das?*

ich: RANDOMSAMPLING und RELAXATION WITH RANDOM ROUNDING (RRRMS).

Juri: *erklären...*

ich: Bei RANDOMSAMPLING wähle ich zufällig eine Belegung der x_i, so dass die Wahrscheinlichkeit, dass eine Klausel der Länge k erfüllt ist, gleich $1 - \frac{1}{2^k}$ ist; gut für lange Klauseln. RRRMS beruht auf linearer Programmierung, wobei die relaxierte Lösung randomisiert gerundet wird.

Juri: *Wie können Sie eine Instanz von* SAT *als lineares Programm ausdrücken?*

ich: (anhand einer Klausel $l_1 \vee l_2 \vee l_3$ erläutert)

Juri: *Wie funktioniert das Runden?*

ich: Nach der Relaxierung erhalte ich für die x_i Werte im Intervall $[0, 1]$, das wird jetzt irgendwie gerundet...

Juri: *Würde es Ihnen helfen, wenn ich Ihnen einen Algorithmus gebe, der eine Zufallszahl im Intervall* $[0, 1]$ *ausgibt?*

ich: Ehrlich gesagt nicht...

Juri: *(hat mir kurz die Lösung gesagt) Können Sie die Analyse des Greedy-Algorithmus für* SC*?*

ich: (Analyse angefangen, $1\frac{1}{2}$ Zeilen geschrieben)

Juri: *Reicht mir. Für den Primzahltest hatten wir im einfachsten Fall den* SIMPLIFIED SOLOVAY-STRASSEN ALGORITHMUS (SSSA) *für jede zweite ungerade Zahl. Worauf beruht der?*

ich: Für n Primzahl ist $a^{\frac{n-1}{2}} \bmod n \in \{-1, 1\}\ \forall a \in \mathbb{Z}_n^*$, d. h. jedes Element ist eulersch. Falls n keine Primzahl sind mindestens die Hälfte der Elemente nicht eulersch.

Juri: *beweisen...*

ich: In der Vorlesung hatten Sie gezeigt, dass die Multiplikation mit einem nicht-eulerschen, invertierbaren Element eine injektive Abbildung von der Menge der eulerschen in die Menge der nicht-eulerschen Zahlen ist. Ich finde den Beweis schöner, dass die eulerschen Zahlen unabhängig von der Voraussetzung $n \equiv 3 \pmod 4$ eine Gruppe bilden, da man aus diesem Beweis mehr schließen kann.

Juri: *Sie meinen, dass der Algorithmus bei Carmichael'schen Zahlen nicht funktioniert?*

ich: Ja. (meinte ich eigentlich nicht, hörte sich aber gut an)

Juri: *Automatentheorie. Die Minimierung von DFA's beruht auf welcher Kongruenz?*

ich: Die Nerode-Kongruenz. (erklärt)

Juri: *All-Aussagen kann man ja algorithmisch schwer prüfen. Wie kann man daraus einen Algorithmus machen?*

ich: (Trennbarkeit von Zuständen erklärt, trennbar mit Länge 0 wenn einer der Zustände Endzustand ist, der andere nicht. Es reicht, Wörter mit endlicher Länge zu betrachten. Markierungsalgorithmus erklärt, besonders auf die Korrektheit hat Herr Hromkovič Wert gelegt, d. h. warum werden alle trennbaren Paare von Zuständen nach endlich vielen Schritten gefunden.)

Juri: *Sie hatten auch einen Zusammenhang zwischen endlichen Automaten und Logik...*

ich: MSO, monadische Logik zweiter Stufe.

Juri: *Was heißt das und wie sieht die Logik genau aus?*

ich: (Quantoren auch über einstelligen Relationen; Wortmodelle erklärt)

Juri: *Wie kann man aus einem Automat ein Formel konstruieren?*

ich: (Konstruktion an einem Beipiel gemacht)

Juri: *Dann warten Sie bitte kurz draußen.*

Fazit: Herr Hromkovič hat irgendwie ein unheimlich gutes Gefühl dafür, was man kann und was nicht. Gerade wenn man denkt, dass man Ihm zu einem Thema fünf Minuten was erzählen kann, unterbricht er einen und geht zum nächsten Thema; wenn man Zeit schinden will, sollte man abschweifen bevor man auf den Punkt kommt. Aber auch wenn man bei einem Thema Probleme hat, versucht er zu helfen. Zum RRRMS hat er gesagt, dass ich doch hätte improvisieren sollen...
Besonders gut hat ihm der Beweis zum SSSA gefallen, da er da mehr hineininterpretiert hat, als ich eigentlich sagen wollte; daher war es insgesamt gerade noch eine 1.0.