PRÜFUNGSPROTOKOLL ZUR THEORETISCHEN INFORMATIK

Fächer .....: Effiziente Algorithmen, Kyrptographie und Compilerbau
Prüfer.....: Prof. Hromkovic
Datum.....: 23.02.2002
Dauer.....: 30 Minuten
Note.....: 1.3

Effiziente Algorithmen:

H (Hromkovic) : Sagen Sie mal was zur dynamischen Programmierung und erklären Sie das Triangulationsproblem.
U (Ich) : Dyn. Prog. ist eine Entwurfstechnik, bei der man mit dem kleinsten Teilproblem anfängt und mit dem nächst größeren Teilproblem weitermacht, bis man das Hauptproblem gelöst hat. Dann habe ich noch das Triangulationsproblem beschrieben und die Beobachtungen erklärt, die in der Vorlesung vorkamen....
H : Wann ist eine Sprache NP-Vollständig.
U : Definition erläutert!
H : Welche Reduktionen kennen Sie?
U : Alle, die wir in der Vorlesung gehabt haben.
H : Dann erklären Sie mal das Hamiltonkreisproblem.
U : Habe zunächst gezeigt, dass HK in der Menge NP liegt. Dann habe ich mit der Konstruktion des Graphen weitergemacht. Als ich mit der Konstruktion fertig war, hat er mich unterbrochen.
H : Kennen Sie lokale Suche?
U : Ja, das ist eine Entwurfstechnik, bei der man zunächst eine Nachbarschaft definiert usw...
H : Wir hatten doch einen Fall, wo lokale Suche ineffizient wird, kennen Sie den Fall?
U : Da wollte er natürlich den patalogischen Fall von TSP hören. Da hat man für eine Eingabeinstanz genau eine optimale Lösung und exponentiell viele zweitbeste Lösungen. Dann habe ich die Diamenten erwähnt.
H : Zeichnen Sie mal wie ein Diamant aussieht.
U : Habe gezeichnet. Dann musste ich noch die Konstruktion erklären und die Kosten der Kanten aufzählen. Das konnte ich leider nicht so gut. Da hat er noch einige Fragen gestellt, an die ich mich jetzt nicht erinnere. Lokale Suche war meine Schwachstelle und genau da hat er nachgehackt.
H : Dann erklären Sie auch mal das metrische TSP.
U : Ich habe den Algorithmus erklärt
H : Können Sie auch beweisen, dass er 2-approximativ ist.
U : Ja, dann habe ich den Beweis vorgeführt.

Kryptographie:

H : Erklären Sie bitte RSA.
U : Man wählt zwei große Primzahlen p,q und.....
H : Können Sie auch beweisen, dass die Entschlüsselung eindeutig ist?

U : Habe die ersten beide Fälle bewiesen, dann hat er nach dem dritten Fall gefragt. Ich habe gesagt, dass dieser Fall in der Realität gar nicht vorkommt und das war wohl auch richtig so.
H : Welche Protokolle kenn Sie?
U : Ich habe alle aufgezählt, die mir gerade einfielen.
H : Erklären Sie mal das Münzwurfprotokoll.
U : A wählt eine Zahl r und verschlüsselt. Dann schickt er diese verschlüsselte Zahl zu B. B rät jetzt, ob r gerade ist oder nicht und teilt das A mit. Dann schickt A zu B die Zahl r und die Entschlüsselungsmethode.
H : Sagen Sie was ein Zero-Knowledge-Proof ist bitte!
U : Habe erklärt.
H : Welche kennen Sie?
U : Alle
H : Dann erklären Sie bitte Graphenisomorphismus.
U : Das habe ich dann auch erklärt. Dann wollte er noch wissen, wieso V aus den Isomorphismen, die er von P bekommen hat, keine geheime Informationen bekommen kann. Da habe ich alles erzählt, was ich dazu wusste. Dann fragte er mich noch, ob ich das auch mathematisch beweisen könne. Da habe ich in die Luft gestarrt und gewartet, bis er etwas sagte. Das war aber nicht schlimm, denn er selber sagte, dass ich das jetzt nicht unbedingt wissen musste. Er wollte nur mal gucken, ob ich das kann.

Compilerbau:

H : Erzählen Sie mal was zur syntaktischen Analyse.
U : Ja, zunächst habe ich die Aufgabe der syntaktischen Analyse erläutert,dann gibt es zwei Methoden, Top-Down und Bottom-Up-Methode. Dann habe ich die LL(k) und LR(k)-Grammatiken erwähnt. Aber ich musste sie nicht aufmalen.
H : Was ist, wenn wir keine Grammatiken wie LL(k) und LR(k) haben?
U : Ich wusste, dass er jetzt den CYK-Algorithmus hören wollte, denn dieselbe frage haben auch zwei meiner Kumpels gestellt bekommen. Da ich aber diesen Algorithmus nicht so gut konnte, habe da gesagt, dass man eine Kontextfreie Grammatik hat und Linksrekursionen beseitigt.
H : Wissen Sie auch wie man Linksrekursionen beseitigt?
U : Ja, man hat direkte und indirekte Linksrekursionen. Indirekte Linksrekursionen beseitigt man dadurch, indem man die Grammatik in eine Greibachnormalform transformiert.
H : Wissen Sie auch, wie eine Greibachnormalform aussieht?
U : Da habe ich die Regeln aufgemalt.

Fazit: Ich habe so gegen Ende vom November angefangen zu lernen. Da ich aber während des Semesters noch mit meinem Seminar beschäftigt war, hat die Vorbereitung auf die Prüfung etwas lange gedauert. Die meiste Zeit habe ich in effiziente Algorithmen und Kryptographie investiert. Für Compilerbau habe ich insgesamt 10 Tage gebraucht. Was ich noch interresant finde, dass ich in Effiziente am Anfang nicht die üblichen einfachen fragen bekamm. Und nachdem ich die Triangulation erklärt hatte, meinte Prof. Hromkovic zu mir, dass ich jetzt schwere Fragen bekommen werde. Dann habe ich gesagt, dass ich mir das vorstellen kann und dass das kein Problem ist. Ich wünsche euch allen noch gute Vorbereitung auf die Prüfung.