Prüfungsprotokoll Diplomprüfung Theoretische Informatik

Pruefer : Prof. J. Hromkovic
Datum : 27.08.2001
Vorlesungen : - Effiziente Algorithmen
- Algorithmische Kryptographie
- Compilerbau
Dauer : ca 35 Minuten
Note : 1.0

Effiziente Algorithmen:
- Wie funktioniert Teile und Herrsche?
- Welche Algorithmen kennen Sie?
- Die Komplexitätsbetrachtung führt auf eine Gleichung, die Rekurrenzgleichung, bekannt als Master-Theorem. Wie lautet es und können Sie es beweisen?

  Habe das Master Theorem hingeschrieben und zu der Formel mit der Summe umgeformt, die man zum Beweise benötigt, er sagte dann, es ist genug.

- Was ist "lokale Suche"?

  Habe das erklärt, das man von einer Eingabe eine (bel.) Lösung erstellt und diese dann lokal ändert, bis man keine bessere Lösung mehr findet. Hier sollte man die Nachbarschaft erwähnen.

- In welchem Fall liefert die lokale Suche immer ein optimales Ergebnis?

  Kam nicht sofort darauf, mit Hilfe von Prof. Hromkowic kam dann "Greedy MST" heraus.

- Gibt es auch Fälle, wo dies nicht geschieht?

  Habe den pathologischen Problemfall für TSP erwähnt und detailliert erklärt.

- Können Sie den beweisen?

  Sollte das informell machen, also mehr erklärend als in Formeln.

- Beweisen Sie das das "metrische TSP" approximativ ist.

  Habe den Algorithmus aus seinem Buch (nicht Christofides Algorithmus) beschrieben und den Beweis schnell erläutert, eher mündlich als schriftlich.

Algorithmische Kryptographie:
- Nennen Sie mir den Unterschied zwischen klassischen und public-key Verfahren.

  Erklärt, das man bei klassischen Verfahren den geheimen Schlüssel aus dem öffentlichen erzeugen kann, bei public-key Verfahren nicht. Public-key Verfahren sind langsamer als klassische Verfahren. Man muss den geheimen Schlüssel über einen sicheren Kanal übertragen, bei public-key Verfahren kann der öffentliche Schlüssel über einen unsicheren Kanal übertragen werden.

- Nennen Sie ein klassisches Verfahren.

  Habe Caeser genannt und beschrieben, habe noch die Namen vieler anderer Verfahren erwähnt (Keyword-Caeser, Hill, Autoclave, Playfair, DES, IDEA, ...)

  Er fragte mich, was Caeser damals noch gemacht hätte, wußte ich nichts zu sagen, er erwähnte, das Caeser zuerst vom Lateinischen ins Griechische übersetze und dann den Shift ausführte.

- Wieviele Schlüssel gibt es bei Caesar?

  Naja 26, die Anzahl der zugelassenen Buchstaben eben.

- Was ist RSA?

  [illegible] die Entschlüsselung funktioniert?

Ja, stand ja nicht umsonst in jedem Prüfungsprotokoll drin, was ich hatte. Nach dem ersten Fall war es genug.

- Nennen Sie ein Protokoll für die digitale Unterschrift.

  Habe das einfachste Protokoll aus der Vorlesung erklärt.

- Was ist das Problem bei diesem Protokoll?

  Ein Lauscher kann mitloggen und sich bei bekannten Paaren (Text, Unterschrift) als unterschreibender Benutzer ausgeben.

- Kennen Sie eine Verbesserung?

  Klar, Unterschrift per El-Gamal.

- Was ist "Zero-Knowledge-Proof"?

  P will V beweisen, daß er ein Geheimnis kennt. P kann V nur mit einer Wahrscheinlichkeit nahe Null betrügen. V weiß hinterher nicht, was das Geheimnis ist. V erfährt nur das, was er aus den bekannten (öffentlichen) Informationen sich selber zusammenreimen kann.

- Welche "Zero-Knowledge-Proofs" kennen Sie?

  Graphen-Isomorphismus, Graphen-nicht-Isomorphismus, ...

- Erläutern Sie einen Isomorphismus?

  Habe Graphen-Isomorphismus erläutert.

- Wie kann man beweisen, das bei ein Zero-Knowledge-Proof der Verifier keine Information erhält?

  Wußte damit nichts anzufangen, er hat's mir dann erklärt. (War wohl nicht wichtig ;-))

Compilerbau:

- Welche Verfahren haben wir für die Syntaxanalyse?

  Natürlich CFG's. BottomUp- oder TopDown-Analyse.

- Nennen Sie Beispiele für TD- und BU-Analyse.

  BU wäre z.B. CYK-Algorithmus. Habe mit dieser Frage nicht viel anfangen können. (War wohl einer weiterführende Frage, die sich nicht auf die Note ausgewirkt hat ;-))
  Er hat dann noch erklärt, das man da ja auch noch LL(k), LR(k) usw. kennt.

- Erklären Sie LL(k).

  Habe die Definition erläutert, er fragte dann, was nun der Knackpunkt wäre. Habe dann erklärt, dass man mit einem k-lookahead entscheiden kann, deterministisch durch tabellengesteuerten Parser.
  Er erwähnte noch, das dies ohne backtracking sei. (War ihm wohl wichtig)

An dieser Stelle wurde ich aufgefordert den Raum zu verlassen.

Alles in allem war die Atmospähre recht angenehm, man wurde zum Teil ermutigt seine Gedanken auszusprechen, andererseits bei technischen Details aber auch schonmal unterbrochen in der Form:
"ok, das reicht, Rest ist ja klar"
Ich habe _nicht_ alles das gewußt, was ich gefragt wurde, einige dieser Stellen habe ich oben angegeben, habe sicherlich ein oder zwei Stellen vergessen.

Als Tipp für Unentschlossene:
Man sollte den Stoff wirklich können (und zwar komplett). Hierbei sind die Idee, aber auch die Details wichtig. Mach die Probeprüfung, die der Lehrstuhl anbietet und arbeite in den entsprechenden Vorlesungen mit, das erleichtert das Verständnis der Theorie enorm.