Fach:	Informatik	**Art der Prüfung:**	Diplomprüfung in Theoretischer Informatik
Prüfer:	Prof. Hromkovic	**Datum:**	09/2005
Dauer:	45 min	**Note:**	1.0
Themen:	Effiziente Algorithmen, Compilerbau, Angewandte Automatentheorie		

Allgemeines

Obwohl Prof. Hromkovic jetzt nur noch in Zürich prüft, habe ich mich entschlossen, doch noch ein Prüfungsprotokoll zu schreiben, da er wohl noch einige Prüfungen in Zürich durchführt und gewisse Dinge doch erwähnt werden sollten (s. abschließende Bemerkungen).

Fragen im Detail

Effiziente Algorithmen

Vorbereitung: Buch von Prof. Hromkovic, Abgrenzung auf der Webseite

- **Frage: Können Sie mir das Konzept der parametrisierten Komplexität erklären?**
  Definition aus Abschnitt 3.3.1, Parameter k erklärt
- **Frage: Wir hatten da zwei Algorithmen. Können Sie die erklären?**
  Die beiden Algorithmen zum Vertex Cover Problem erklärt. Wichtig: Die drei Beobachtungen sind das Hilfsmittel zum Erklären der Algorithmen, z.B. wollte er hören, dass aufgrund der ersten Beobachtung nur noch Knoten mit Grad $\leq k$ in G' sind und aufgrund der zweiten nur noch $m(k+1)$ Knoten nach einem VC der Größe m zu durchsuchen sind.
- **Frage: Laufzeit der Algorithmen?**
  Die Angabe der Laufzeit war ausreichend, keine Herleitung
- **Frage: Können sie mir erklären, was man unter stark NP-schwer versteht?**
  Zuerst habe ich die Folgerung für einen Pseudo-Polynomzeit-Algorithmen genannt, nämlich dass es bei einem stark NP-schweren Zahlproblem keinen gibt. Dann die Definition aus Abschnitt 3.2.4 genannt.
- **Frage: Kennen Sie denn ein stark NP-schweres Problem?**
  TSP
- **Frage: Können Sie beweisen, dass es stark NP-schwer ist?**
  Dazu muss man $HC \leq_p Lang_{Value(p)-TSP}$ zeigen, da es sich beim TSP um ein Optimierungsproblem handelt und sich für die Reduktion HC anbietet (s. Abschnitt 3.2.4). Der Beweis sollte sitzen, er ist auch nicht schwierig.
- **Frage: Was ist ein PTAS (Polynomzeit-Approximationsschema)?**
  Definition aus Abschnitt 4.2.1: ... $\varepsilon_A(x) \leq \varepsilon$ und $Time_A(x, \varepsilon^{-1})$ polynomial beschränkt in $|x|$
- **Frage: Gibt es auch etwas, dass noch irgendwie in ε beschränkt ist?**
  FPTAS: $Time_A(x, \varepsilon^{-1})$ auch polynomial in ε^{-1} beschränkt
- **Frage: Wir hatten da auch einen Algorithmus? Können sie den erklären?**
  PTAS für SKP erklärt, wichtig war, dass alle k-elementigen Teilmengen gebildet und dann, wie im Algorithmus davor, um Elemente erweitert werden und dass $k = \lceil \frac{1}{\varepsilon} \rceil$.
- **Frage: Jetzt noch die Approximationsgüte?**
  Hier musste ich den Beweis nachvollziehen und zwar so weit, bis der Rest nur noch einsetzen und umformulieren war, das musste ich dann nicht mehr machen.

Compilerbau

Vorbereitung: Mitschrift SS 2005, Vorlesungsvideos von http://video.s-inf.de

- **Frage: Was macht man bei der syntaktischen Analyse?**
  Zerlegung der Symbolfolge in syntaktische Einheiten
- **Frage: Was für Verfahren gibt es da?**
  CYK-Algorithmus, Top-Down-Analyse, Bottom-Up-Analyse
- **Frage: Welche Grammatiken braut man beim CYK-Algorithmus und wie ist die Laufzeit?**
  Kontextfreie Grammatiken, $O(n^3)$ Zeit und $O(n^2)$ Platz
- **Frage: Und welche Laufzeit streben wir an? Welche Verfahren gibt es denn da so und welche Grammatiken braucht man dafür?**
  Top-Down-Analyse mit LL(k)-Grammatiken und Bottom-Up-Analyse mit LR(k)-Grammatiken
- **Frage: Definition von LL(k)-Grammatiken?**
  s. Skript

Angewandte Automatentheorie

Vorbereitung: Vorlesungsvideos vom Lehrstuhl, Skript vom Lehrstuhl

- **Frage: Können Sie mir etwas zur Minimierung von DEA sagen?**
  Minimierung basiert auf kanonischer Zustandsäquivalenz, der damit reduzierte Automat ist isomorph zum kanonischen DEA (Äquivalenzklassenautomat bzgl. der Nerode-Äquivalenz), der minimal ist, also: Reduzierung liefert Minimalautomat, Durchführung mit Blockverfeinerungs-Algorithmus bzw. Markierungsalgorithmus
- **Frage: Und geht das bei NEA auch so einfach?**
  Nein, das NEA-Minimierungsproblem ist PSPACE-schwer.
- **Frage: Wie beweist man das?**
  Hier musste ich die Beweisidee aus dem Skript erklären: Universalitätsproblem ist spezielles Minimierungsproblem, also: zeigen, dass Nicht-Universalitätsproblem PSPACE-schwer ist mit Hilfe einer polynomial-platzbeschränkten Turingmaschine.

Abschließende Bemerkungen zur Prüfung

Ein paar wichtige Punkte zur Prüfung:

1. Beim Lesen der Prüfungsprotokolle könnte man den Eindruck bekommen, Prof. Hromkovic vergibt nur Einser-Noten. Dem ist definitiv nicht so, was mir einige meiner Mitstudenten bestätigen konnten. Es ist halt nur eben so, dass diejenigen mit 3.x und 4.0 keine Prüfungsprotokolle schreiben, weil man nach so einer Note oft keine Lust hat, eines zu schreiben, erst recht nicht, wenn alle existierenden Prüfungsprotokolle Einser-Noten tragen.
2. Wenn man die Grundlagen (z.B. Def. von NP-schwer/NP-vollständig, O-Notation sowie praktisch alles aus dem Kapitel 2.3 seines Buches) nicht beherrscht, hat man keine Aussicht mehr auf Bestehen der Prüfung mit einer guten Note. Hat man Probleme mit den "schwierigen" Sachen (sprich: den Fragen, die man so aus den Einser-Prüfungsprotokollen kennt), fragt Prof. Hromkovic gerne mal nach den Grundlagen und dann müssen die sitzen.
3. Was schon in allen früheren Prüfungsprotokollen steht, kann ich auch nur betonen: Die Beweise müssen alle sitzen. Auch die, die man am liebsten überlesen würde. ;-)