Prüfungsprotkoll

Prüfer: Professor Hromkovič

Fächer:

- Algorithmische Kryptographie (Unger)
- Approximative und Randomisierte Algorithmen (Hromkovič)
- Compilerbau (Indermark)

Datum: 18.02.2002

Note: 1.0

Dauer: 40–45 Minuten

- Die Reihenfolge der Fächer konnte ich mir aussuchen. Überhaupt war die gesamte Athmosphäre sehr locker, was die eigene Nervosität doch erheblich gemindert hat.
- Leider ist mein Kurzzeitgedächnis etwas schwach, deswegen ist dieses Protokoll vermutlich nicht ganz vollständig und auch nicht so ausführlich wie einige der anderen Protokolle.

Fach 1: Approximative und Randomisierte Algorithmen

Klassifizierung von randomisierten Algorithmen (Las Vegas und die verschiedenen Monte Carlo)

Beispiele für Monte Carlo–Algorithmen (NonEq / Primzahltest / NonEq–Pol / 1BP)

Idee und Ablauf des NonEq–Pol Algorithmus (Fingerprinting / Vergleich der zuf. Belegung der Polynome modulo n)

Beweis der Wahrscheinlichkeit $(m * d)/n$ für NonEq–Pol

Idee und Ablauf von MinCut–Algorithmus (zunächst der einfache Random Contraction)

Beweis der Erfolgswahrscheinlichkeit des Algorithmus (Wahrscheinlichkeit minimalen Schnitt zu finden: $2/(n * (n-1))$

Idee der Verbesserungen von MinCut mit Begründung (nach einigen Contractions deterministisch / rekursiv mit mehreren randomisierten)

Rucksackproblem: nenne FPTAS für KP (F–KP samt Idee, basiert auf DPKP, Skalierung der Eingabegröße)

Nachfrage: um welchen Faktor skalieren ($d_i := c_i * 2^{-t}$ mit $t := log_2((\epsilon * c_{max})/((1+\epsilon) * n)$

Nachfrage: Idee von DPKP (Dynamische Programmierung, besten Praefix finden und verlängern)

Was ist GAP–Reduktion? (Methode um Inapproximierbarkeit zu zeigen, weitere Methoden: NP– / AP–Reduktion, PCP–Satz)

Idee von GAP–Reduktion (konnte ich nicht beantworten)

Beispiel für NP–Reduktion? (Reduktion TSP auf Hamiltonkreis mit Beschreibung und grober Begründung)

Fach 2: Kryptographie

Aufbau RSA erklären (durchgeführt)

Beweis für Eindeutigkeit (nachdem ich kurz die verschidenen Fälle erwähnt hatte und die Idee des Beweises per chin. Restklassensatz und Satz von Fermat angedeutet hatte hat Prof. Hromkovič unterbrochen)

Liste der Protokolle, die auf Public–Key Systemen basieren geben (durchgeführt, sind fast alle nach etwa zehn unterbrochen worden)

Oblivious Transfer erklären (beide Fälle erklärt, sollte dann Protokoll für den ersten Fall, 50/50 beschreiben)

Fach 3: Compilerbau

Was ist ein Compiler? (Standartantwort: Compiler ist ein Programm zur Übersetzung von ..., dann sofort Phasenaufbau relativ ausführlich erklärt)

Nachfrage lexikalische Analyse (Reguläre Ausdrücke erkannt durch NFA, Pot.mengen–Konstruktion von DFA)

Nachfrage syntaktische Analyse, welche Methoden (CYK–Algorithmus zur Erkennung allgemeiner CFG, Komplexität $O(n^3)$, Platzkompl. $O(n^2)$, hier wurde ich unterbrochen)

Nachfrage zu CYK, Idee, welche Grammatiken (Idee: dynamische Programmierung, Grammatiken in Greibach Normalform, mehr konnte ich nicht sagen, war aber auch nicht schlimm, Prof. Hromkovič war auch der Meinung, dass das nicht direkt zu Compilerbau gehört)

jetzt aber: Nachfrage LL(k), LR(k) (erklärt: dient zur Beseitigung von Nichtdeterminismus, benutzt in Top–Down sowie Bottom–Up Analyse, hier unterbrochen)

Nachfrage: kann jede Sprache so erkannt werden (nein, da insbesondere nicht jede Sprache, die durch CFG beschrieben werden kann deterministisch erkannt werden kann)

jetzt Definition von LL(k), LR(k) (LL(k) aufgezeichnet, bei LR(k) in den letzten Worten unterborchen, ich war entlassen.

Ich sollte an dieser Stelle den Raum verlassen, wurde aber bereits wenige Augenblicke später wieder hereingebeten, mir wurde sofort mitgeteilt, dass ich eine 1.0 erreicht habe. Zur Begründung hat Prof. Hromkovič gesagt, dass ich lediglich eine Frage aus AuRA nicht beantworten konnte, da AuRA aber eine Vertiefungsvorlesung sei wäre das vertretbar.