Prüfungsprotokoll

Prüfer: Professor Hromkovič (H), Dr. Walter Unger (U)

Fächer:

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

Datum: 18.02.2002

Note: 1.0

Dauer: 45 Minuten

H: Guten Tag... Womit wollen sie denn anfangen?

T: Ich würde gerne mit Crypto beginnen und mit LP aufhören.

H: OK... kommen wir zu Crypto.. übergibt an Walter.

Fach 1: Kryptographie

W: Erklären Sie RSA.

T: RSA erklärt und direkt gesagt das man die Eindeutigkeit beweisen kann.

W: Ja und wie?

T: Dann habe ich die Eindeutigkeit bewiesen und dabei vergessen zu erwähnen dass $w < n$ sein muss. Das habe ich aber auf einen kurzen Hinweis hin sofort nachgeholt.

W: Welche Probleme eignen sich denn noch für Public Key Systeme?

T: Aufgezählt und immer direkt gesagt welche, wenn es da einen Namen für gibt: Diskrete Logarithmus : El Gamal, Elliptische Kurven, Isomorphismen, alle Probleme aus NP.

W: Kennen Sie El Gamal?

T: Ja (obwohl es eigentlich nicht in der Vorlesung drin war), so: dann erklärt wie El Gamal geht und direkt auch die Entschlüsselung bewiesen.

W: Welche Protokolle hatten wir denn?

T: Alle aufgezählt die wir hatten (oder hatte ich doch eins vergessen), wusste Walter aber auch nicht sicher, ob ich alle aufgezählt habe. War aber wohl OK so.

W: Bei welchen Protokollen kommen denn die quadratischen Reste vor?

T: Da fielen mir Münzwurf und Lockable Box zu ein.

W: Da gibt es noch eins (Walter und Prof. Hromkovič grinsen).

T: Mmmh... grübel... ne, keine Ahnung.

W: Oblivious Transfer.

T: Ach ja... dann habe ich mal grob erklärt, was Oblivious Transfer ist, und wie dieser geht.

W: Was fällt Dir zu Zero-Knowledge Proofs ein? (Musste ja kommen!)

T: Dann habe ich mal so erklärt was ein Zero-Knowledge-Proof ist, welche Bedingungen da gelten müssen und gesagt für welche Probleme man einen Zero-Knowledge Proof machen kann: Alle aus NP mit positiver Antwort und noch einige andere wie z.B. Graphisomorphie.

W: In welcher Klasse liegt denn Graphisomorphie?

T: Dazu habe ich gesagt, dass ich das nicht genau wüsste aber vermute, dass es in der Klasse ist, die NP und P trennt.

W: Kannst du den Zero-Knowledge Proof fürs Bandweiten-Problem *grins*?

T: Mmh... nein.

H: Man könnte die Reduzierung machen.

T: Das haben wir damals in der Übung so gemacht und 0 Punkte dafür bekommen.

W: *grins* Genau das wird auch vermutet. Damit bin ich sehr zufrieden und übergibt wieder an Prof. Hromkovič.

Fach 2:Approximative und Randomisierte Algorithmen

H: Welche Klassen von Randomisierten Problemen kennen sie?

T: Habe Las Vegas und Monte Carlo erklärt und erwähnt, dass es auch noch randomisierte Approximierungsalgorithmen gibt.

H: Schön. Was ist denn der Unterschied zwischen Monte Carlo mit beschränktem und unbeschränktem Fehler?

T: Habe erklärt, dass man bei unbeschränktem Fehler exponentiell oft wiederholen muss um an eine gewünschte Fehlerwahrscheinlichkeit ranzukommen. Das war scheinbar nicht ganz tiefschürfend genug, denn da wurde zwischendurch immer wieder mal nachgehakt. Und nach Gründen gefragt.

H: Dann suchen wir mal ein Beispiel. Erklären Sie mir den Algorithmus zum Vergleich zweier Polynome.

T: (Mist... das kann ich nicht so richtig) Habe das Prinzip erläutert und gesagt was für Wahrscheinlichkeiten da rauskommen.

H: Können Sie das beweisen.

T: (Mist.. das kann ich gar nicht) Habe dann die Beweisskizze erklärt und gesagt das macht man mit Induktion aber das war offensichtlich nicht genug :-(

H: Wollen Sie jetzt hier an diesem Problem noch weiter machen oder sollen wir uns ein anderes suchen?

T: Lieber ein anderes.

H: Gut. Gehen wir zum Simplified Solovay Strassen Algorithmus. Ich gehe davon aus, dass sie den Satz von Fermat beweisen können. Beweisen Sie die Aussage über die Anzahl der Zeugen

T: Kurzüberblick über den gesamten Algorithmus gegeben und dann den Beweis gemacht. Das war wohl ziemlich gut, denn da wurde nicht nachgehakt.

H: Kommen wir zu Approximativen Algorithmen..

T: Da habe ich dann einfach mal dazwischen gelabert und einen Überblick über diesen Teil der Vorlesung gegeben: Rucksackproblem, TSP, Inapproximierbarkeit.

H: Erklären Sie mir bitte was ein Approximationsschema ist.

T: Definition erklärt und direkt gesagt dass es noch echte Approximationsschemata gibt.

H: Gut. Gehen wir zum Rucksackproblem. Da hatten wir ein PTAS für das einfache Rucksackproblem.

T: Algorithmus erklärt, und Beweis angerissen. Dabei bin ich unterbrochen worden, als ich bei $w_{i_q} < b/(k+1)$ angekommen war. Das reichte ihm dann.

H: Gut. Am Ende der Vorlesung hatten wir noch einige Sachen über Inapproximierbarkeit.

T: Da gibt es dreieinhalb oder vier Verfahren: Reduktion auf NP-harte Entscheidungsprobleme, APX-Reduzierung, GP-Reduzierung und PCP-Satz. Zu jedem immer direkt gesagt was wir in der Vorlesung damit gemacht haben. Und was es vom Konzept her bedeutet.

H: *grins* Gut. Können Sie mir die GP-Reduktion von MAX-E3-SAT auf MAX-2SAT beweisen?

T: Rumgelabert wie das funktioniert, aber beweisen kann ich es nicht.

H: Dann hatten wir ja auch noch randomisierte Approximierungsalgorithmen. Können Sie mir erklären welche beiden Algorithmen wir für MAX-SAT verwendet und kombiniert haben?

T: Klar, RSMS und RRRMS kombiniert ergibt 4/3-approximativen Algorithmus und beide Algorithmen erklärt.

H: Gut. Das reicht mir.

Fach 3: Logikprogrammierung

H: Erklären Sie mir Resolution und Unifikation.

T: Dazu habe ich dann einen kleinen Überblick über die gesamte Logikprogrammierung gegeben, der vom Inhalt her ungefähr folgendes war:

- Resolution ist ein syntaktisches Verfahren um Unerfüllbarkeit zu zeigen.
- Kommt aus der Aussagenlogik.
- In der Logikprogrammierung wird sie auf die Prädikatenlogik erweitert.
- Dazu benötigt man die Unifikation.
- Definition Unifikator/allgemeinster Unifikator

- Unerfüllbarkeit ist interessant, weil $\Phi \models \psi \Leftrightarrow \Phi \cup \not\psi$ unerfüllbar.
- Resolution funktioniert folgendermaßen...
- In der Logikprogrammierung verwendet man die Hornlogik, die eine echte Einschränkung darstellt, in der Praxis aber in der Regel vollkommen ausreicht, und der Vorteil davon ist, dass die Unerfüllbarkeit sich in Linearzeit testen lässt.

H: Gut. Damit haben Sie mir direkt meine nächste Frage vorweggenommen. Sie haben ja im Speziellen Prolog gemacht. Da könnten Sie ein Beispiel machen. Suchen Sie sich eins aus.

T: Ich habe mir den Nichtdeterministischen Automaten ausgesucht, weil ich wusste, dass er das schon-einmal gefragt hatte. Und dementsprechend konnte ich das wirklich sehr gut. Dabei habe ich das auch alles immer erläutert.

H: Gut. Können Sie mir die Vor- und Nachteile von Prolog nennen?

T: *grübel* ineffizient (damit war er nicht wirklich zufrieden. Er grummelte irgendwas von wegen das könne man schlecht abschätzen und man habe keinen Einfluss auf die Effizienz) und dann sagte ich noch es sei eine deklarative Programmiersprache bei der man das Problem und nicht die Lösung beschreiben muss. Auch erwähnt, dass in Prolog das deklarative Konzept wegen Tiefensuche und Cut nicht 100% eingehalten wird. Auch kann man damit Rekursive Anfragen stellen (darüber hat er sich gefreut).

H: Gut. Und wo wird Prolog angewendet?

T: z.B. in Datenbanken... Datalog erwähnt und gesagt dass Datalog Breitensuche verwendet, weil in Datenbanken alle Lösungen interessant sind, bei Prolog jedoch normalerweise immer nur eine. Und dann noch erwähnt, dass es in Datalog keine Funktionen gibt.

H: Gut. Das reicht mir. Bitte lassen Sie uns kurz allein.

Etwas später durften wir (ich und meine zwei Zuschauer) wieder reinkommen und er hat gesagt dass ich eine 1.0 bekommen hätte, und erläutert wie es zu dieser Note kam:

1. Man konnte sehr gut sehen, dass ich nicht nur die einzelnen Teile beherrsche sondern auch Verständnis von “oben bis unten” habe.
2. Crypto war hervorragend.
3. Bei AuRA gab es zwar eine kleine Schwäche, dies ist bei AuRA aber verzeihlich, da es eigentlich eine Vertiefungsvorlesung sei und man da einen “Joker” habe.
4. Logikprogrammierung eigentlich nicht sein Fach sei, er aber den Eindruck habe, dass ich dort gut vorbereitet sei und einen Überblick habe.

Auch sei es bei Prüfungen sehr wichtig nicht nur die Einzelteile zu beherrschen sondern auch den Gesamtzusammenhang zu verstehen.

Fazit

Ich empfand die Prüfung als sehr angenehm. Zwei Freunde hatten gefragt, ob sie zuschauen dürften, dem habe ich zugestimmt, und mich hat dies auch nicht gestört, denn meine Nervosität ist verflogen, sobald ich einmal angefangen hatte. Euch wünsche ich viel Glück und empfehle: Schaut euch mal eine Prüfung an, schreibt selber Prüfungsprotokolle, lernt fleißig. Wie bereits gesagt: Fast mehr als die nackten Fakten zählen Zusammenhang und Überblick.