Prüfungsprotokoll Theoretische Informatik

Ewgenij Sokolovski

26. November 2005
Last updated am 29. November 2005

Ich bedanke mich sehr bei all denen, die ihre Prüfungsprotokolle und Zusammenfassungen von Prüfungsfragen ins Netz gestellt haben. Diese Unterlagen haben mir SEHR bei der Vorbereitung zu meiner eigenen Prüfung geholfen!

Zusammenfassung

Das ist ein Protokoll über meine Diplomprüfung im Fach Theoretische Informatik, die am 23.11.05 um 9:00 stattgefunden hat. Prüfer waren Professor Klaus Indermark und Dr. Walter Unger. Geprüft wurden die Fächer Algorithmische Kryptografie (Unger), Effiziente Algorithmen (Unger) sowie Logikprogrammierung (Indermark). Die Prüfung wurde mit der Note 1.0 bewertet. Die Dauer der Prüfung betrug fast exakt 45 Minuten.

Der Stoff war:

- Algorithmische Kryptografie - Vorlesung von Dr. Unger (Skript dazu im Netz)
- Effiziente Algorithmen - Vorlesung von Professor Hromkovic im Wintersemester 2003/04, hab mich aber ausschließlich nach dem Buch für die Prüfung vorbereitet
- Logikprogrammierung - Vorlesung von Professor Indermark im Wintersemester 2003/04, vorbereitet dann nach den im Netz verfügbaren Skripten von 2001 und 2004.

Die Prüfung hat im Büro von Professor Indermark stattgefunden. Ich war schon um 8:40 da, recht nervös. Prof. Indermark saß schon in seinem Büro. Etwa um 8:55 kam Dr. Unger und die Prüfung begann.

1 Algorithmische Kryptografie

- **Indermark:** In welcher Reihenfolge möchten Sie die Fächer geprüft haben?
- **Ich:** Kryptografie, Effiziente Algorithmen, Logikprogrammierung
- **Indermark:** Also Logik ganz am Ende, OK (übergibt an Dr.Unger)
- **Unger:** OK, fangen wir mit etwas Einfachem an: Rabin-Verschlüsselung.
- **Ich:** Rabin-Verschlüsselung und Entschlüsselung erklärt. Erwähnt, dass die Zahlen p,q $\equiv$ 3 mod 4 sein sollten, um die Entschlüsselung zu beschleunigen. Habe auch gesagt, dass die Kryptoanalyse beweisbar so schwer ist wie die Faktorisierung von n = p*q. Zu den Einzelheiten der Wurzelberechnung und wie man denn von den vier möglichen Wurzeln den Plaintext bestimmt, wollte er nichts wissen.
- **Ich:** Also widerspricht das Brechen von Rabin der Diffie-Hellmann-Vermutung.
- **Unger:** (große Augen) Wie bitte???
- **Ich:** (Oh, scheiße, wenn ich schon direkt zu Beginn der Prüfung sowas von mir gebe...) Ach ja, sorry, Quatsch, Diffie-Hellmann ist die Vermutung über den diskreten Logarithmus, hat mit Rabin nichts zu tun... **Unger:** Sondern mit El Gamal **Ich:** Ich meinte natürlich die Vermutung, dass man die quadratische Wurzel nicht effizient berechnen kann.
- **Unger:** Kannst du auch beweisen, dass die erfolgreiche Kryptoanalyse bei Rabin äquivalent zum Faktorisieren von n ist?
- **Ich:** Ja. (Hab ihm das bewiesen. Die einfache Richtung, dass wenn man n faktorisiert, man auch die quadratische Wurzel ziehen kann. Und auch die Trickige, dass wenn man quadratische Wurzel ziehen kann, man auch n faktorisieren kann. Sie steht im Buch von Delfs-Knebl auf der Seite 266 Lemma A.58. Scheint eine der Lieblingsfragen von Unger zu sein, also würde ich auf jeden Fall lernen.)
- **Unger:** So, wie gut ist denn RSA?
- **Ich:** Es wird vermutet, dass RSA ebenfalls genau so schwer zu brechen ist, wie n zu faktorisieren, allerdings gibt es noch keinen Beweis dafür.

- **Unger:** Was kannst du denn noch über den Plaintext bei RSA sagen?
- **Ich:** Meinen Sie jetzt den Eindeutigkeitsbeweis?
- **Unger:** Nein, nein. Da war was mit den Bits.
- **Ich:** Ah ja, wenn man einen Orakel hätte, der bei einem gegebenen Cryptotext uns immer den letzten Bit von dem entsprechenden Plaintext sagen könnte, dann könnten wir sukzessive RSA brechen.
- **Unger:** Kannst du das beweisen?
- **Ich:** Nein, leider nicht.
- **Unger:** Hast du denn eine Idee, wie man das machen könnte? Da gab es zwei Verfahren, nenn mir den einfachen.
- **Ich:** Nee, weiss ich nicht:((
- **Unger:** OK, ist nicht so schlimm. Was heisst denn das alles für die Sicherheit von RSA?
- **Ich:** Das heisst, dass RSA ist genau so sicher wie seine Teilinformationen.
- **Unger:** OK. Gut. Was kannst du mir zu Zero-Knowledge erzählen?
- **Ich:** Vic erfährt nichts von Peggy, was er nicht selbst hätte berechnen können. Habe dann das mit dem probablistischen Simulator erklärt, dass er dann nach polynomiell vielen Versuchen in der gemeinsamen Eingabe x einen akzeptierenden Transkript ausgibt.
- **Unger:** Und was kann man über diese Transkripte sagen?
- **Ich:** ??? Was meinen Sie jetzt?
- **Unger:** Ja, was kann man über diese Transcripte im allgemeinen sagen?
- **Ich:** Ach ja, sie sind statistisch genau so verteilt wie die Transcripte von dem wirklichen Beweissystem.
- **Unger:** Genau. Und warum muss man dem probablistischen Simulator unbedingt einen Vic* als Eingabe geben und nicht einfach Vic? (Vic* ist der allgemeine Verifier, also auch ein unter Umständen unehrlicher, und Vic ist der ehrliche Verifier)

- **Ich:** Ja, weil der Begriff Zero-Knowledge impliziert, dass der Verifier keine neuen Informationen kriegt, egal was er macht. Ob er nun ehrlich ist oder nicht.
- **Unger:** Was wäre denn, wenn man anstatt Vic* einfach Vic schreiben würde?
- **Ich:** Dann wäre nur der Fall berücksichtigt, dass der Verifier ehrlich ist. Ein unehrlicher könnte also eventuell zusätzliche Informationen für sich gewinnen.
- **Unger:** Das ist nicht ganz das, was ich meine. Was wäre denn mit den Transkripts, die dadurch produziert worden wären? Wie sähen sie aus?
- **Ich:** Ahso, die Herausforderungen - die $e \in \{0,1\}$ wären nicht mehr zufällig.
- **Unger:** Ja, genau! (Anscheinend wollte er dieses „zufällig" unbedingt hören)
- **Unger:** Welche Zero-Knowledge-Protokolle kennst du?
- **Ich:** Fiat-Shamir, Graphisomorphismus, Graphnichtisomorphismus, 3-Färbbarkeit
- **Unger:** Tjaa, dann erklär mir mal das Protokoll für Hamiltonkreis.
- **Ich:** (Mist, warum hat er denn nicht eines der von mir genannten gefragt? Hamiltonkreis, hmm, hmm. Habe ich irgendwann gelernt, aber schon was länger her, denn es tauchte in keinem der Prüfungsprotokolle auf. Na gut, versuchen wir es mal, zu verlieren hab ich ja nix:))) Hab es dann doch geschafft, das Protokoll zu erklären, er ging allerdings nicht sehr auf die Details ein, sonst wäre ich wahrscheinlich dran gewesen.
- **Unger:** Kennst du das Geburtstagsprotokoll?
- **Ich:** Ja, klar. (Habe das Protokoll ausführlich, also jeden Schritt beschrieben. Als ich bei der Addition von 1 zu den $z's$ kam, hat er mich abgebrochen und gesagt, dass der Rest wohl klar wäre)
- **Unger:** Jetzt mal eine ganz provokative Frage: Ist denn dieses Protokoll Zero-Knowledge?

- **Ich:** Hmm, Hmm (überleg, überleg). Also eigentlich scho... Ach nee, ist er nicht. Die Partner erfahren ja dabei, wer von Ihnen älter ist, d.h. sie erfahren etwas, was sie ohne einander nicht hätten berechnen können. Und das wiederum widerspricht den Zero-Knowledge Anforderungen. (Hab das in der Prüfung nicht so ausführlich gesagt, aber mich halt verständlich ausgedrückt.)
- **Unger:** Genau!!!
- **Unger:** Was kannst du mir jetzt zu den Wahlen sagen?
- **Ich:** (Hurra, hab das noch am Abend davor wiederholt:)) Zum Beispiel gibt es da eine Möglichkeit, mit den homomorphen Commitments ein Wahlsystem für Ja/Nein Wahlen aufzusetzen.
- **Unger:** Ja, und wie genau?
- **Ich:** Hab ihm dann zuerst ausführlich erklärt, was homomorphe Commitments sind und dann das Protokoll zum Wählen damit.
- **Unger:** OK, und wenn man jetzt dieses v^s hat, wie berechnet man s? Dafür müsste man doch den diskreten Logarithmus berechnen.
- **Ich:** Nein, man löst das durch einfaches Probieren. Es gibt ja nicht sooo viele Menschen auf der Erde, dass man es nicht hinkriegen würde. Auch wenn alle 6 Milliarden an einer Abstimmung teilnehmen, kann man einfach für s nacheinander Zahlen von 1 bis ... einsetzen, $v^1, v^2, ...$ berechnen und mit dem Ergebnis vergleichen.
- **Unger:** Ja, das könnte man. Wie kann man das denn noch schneller machen?
- **Ich:** ???
- **Unger:** Ja, wie lange würder der Wahlrechner brauchen, um das Ergebnis durch Probieren zu berechnen, wenn man 6 Milliarden Wähler hätte?
- **Ich:** (unsicher)Ein Paar Sekunden.
- **Indermark:** (da mischt sich der Indermark ein) Nee, also mit ein paar Sekunden wären sie nicht ausgekommen.
- **Ich:** Ein paar Minuten, Stunden?

- **Indermark:** Oder Tage.
- **Unger:** Also wie könnte man das beschleunigen?
- **Ich:** Öööhhmm...
- **Unger:** OK, ich sags dir. Du lässt einfach jeden Wähler von diesen 6 Milliarden ein v^s ausprobieren:))
- **Ich:** Aaaahhhhh. (Jetzt ist der Groschen gefallen:)))
- **Unger:** OK, Effiziente Algorithmen.

2 Effiziente Algorithmen

- **Unger:** Was kannst du mir über die approximativen Algorithmen erzählen?
- **Ich:** Blablabla - Definition gesagt, Approximationsgüte und Approximationsfehler erklärt, PTAS und FPTAS erwähnt.
- **Ich:** Soll ich jetzt PTAS und FPTAS erklären?
- **Unger:** Ja, das wäre die nächste Frage.
- **Ich:** Blablabla - die Definitionen aufgesagt
- **Unger:** Gut. Welche approximative Algorithmen kennst du?
- **Ich:** Den für SKP, den einfachen für $\triangle-$TSP, den PTAS für das allgemeine KP, den FPTAS für das allgemeine KP, den Christofides Algorithmus für $\triangle-$TSP.
- **Unger:** Dann erklär mal den Algorithmus für das Rucksackproblem.
- **Ich:** Den Einfachen oder den PTAS?
- **Unger:** Den Einfachen.
- **Ich:** Hab den Algorithmus erklärt und die Güte davon gesagt.
- **Unger:** Kannst du diese Güte beweisen?
- **Ich:** Ja. Habe dann angefangen, zu beweisen, Unger hat mich aber schon nach der Formel $w_{j+1} \leq w_j \leq \frac{w_1+w_2+\cdots+w_j}{j} \leq \frac{b}{j}$ unterbrochen: „Ich sehe schon, dass es dann aufgeht“

- **Unger:** So, jetzt nehmen wir mal an, der Christofides Algorithmus wäre schon erklärt. Kannst du mir seine Güte von 1.5 beweisen?
- **Ich:** Ja. Habe dann den Beweis angefangen: Blablabla und da die Anzahl der Knoten mit dem ungeraden Grad immer gerade ist...
- **Unger:** Und warum ist sie gerade?
- **Ich:** (Mist, jetzt das noch, keine Ahnung) Es gibt da einen Satz, der das besagt.
- **Unger:** Kannst du den auch beweisen (grinst).
- **Ich:** Ähhhh, nein, kann ich nicht.
- **Unger:** Das ist aber ein Satz aus dem Grundstudium (grinst)
- **Ich:** Nee, trotzdem nicht:))
- **Unger:** OK, ist nicht so schlimm, mach weiter.
- **Ich:** Blablabla, Unger hat mich aber wiederum abgebrochen(„Ich sehe, dass es dann aufgeht“) als ich die zwei Matchings M_1 und M_2 sowie $cost(M) \leq min\{cost(M_1), cost(M_2)\}$ erklärt habe. (Dass daraus $cost(M) \leq \frac{1}{2} cost(H_{Opt})$ folgt, musste ich nicht mehr zeigen)
- **Unger:** So, dann hatten wir noch die Diamanten. Wobei braucht man sie?
- **Ich:** Bei dem Pathologischen TSP und bei der Reduktion $HC \leq_p RHC$.
- **Unger:** Dann erklär mal $HC \leq_p RHC$.
- **Ich:** Blablabla - als ich die Abbildung von Knotenverbindungen in G auf Diamantenverbindungen in G' erklärt habe, hat Unger mich unterbrochen.
- **Unger:** Erklär mir jetzt mal die Idee vom Schöning Algorithmus.
- **Ich:** Hab ihm dann die Idee erklärt: „Man nehme so und so viele Mal (die genaue Zahl war ihm total unwichtig) eine beliebige Lösung und mache 3*n Schritte der localen Suche. Die Wahrscheinlichkeit, dass man dann doch keine Lösung findet, obwohl sie exisitiert ist sehr klein (wie klein wollte er auch nicht wissen)“
- **Unger:** OK. Ich bin fertig. (Übergibt an Professor Indermark).

3 Logikprogrammierung

- **Indermark:** So, dann fangen wir an. Warum kann man überhaupt in Logik programmieren?
- **Ich:** ??? Öhhhm, ja, weil man das Problem in logischen Formeln, in den Hornklausel ausdrücken kann.
- **Indermark:** Na ja, ein Problem kann ich auch in natürlicher Sprache ausdrücken.
- **Ich:** Ja, aber in Hornlogik gibt es einen effizienten Erfüllbarkeitstest. Man kann dann mit Hilfe der Resolution Probleme lösen.
- **Indermark:** Aha! Resolution. (Wollte anscheinend auf sowas hinaus)
- **Indermark:** Erklären Sie doch die Folgerungsbeziehung und ihre Bedeutung für Logikprogrammierung.
- **Ich:** Bedeutung von $\Phi \models \varphi$ erklärt und die Äquivalenz

  $$\Phi \models \varphi \Leftrightarrow \Phi \cup \neg\varphi \text{ unerfüllbar}$$

  erläutert. Auch erklärt, dass dieses $\neg\varphi$ die Anfrage an das Logikprogramm ist.
- **Indermark:** Wie viele Modelle kann es denn zu einer Formel geben?
- **Ich:** Unendlich viele.
- **Indermark:** (Guckt mich verdutzt an) Unendlich viele??
- **Ich:** Ja, es gibt ja unendlich viele Strukturen, die diese Formel in FO erfüllen können.
- **Indermark:** Ich meinte die Aussagenlogik, nicht die Prädikatenlogik.
- **Ich:** Ahsoooo, da gibt es nur endlich viele Modelle.
- **Indermark:** Wie schnell kann man denn den Erfüllbarkeitstest in der Aussagenlogik durchführen?

  Und jetzt Achtung! Jetzt kommt die wahrscheinlich größte Prüfungsblamage der letzten 10 Jahre:)))

- **Ich:** In $O(n)$ (Keine Ahnung, was für ein Viech mich gebissen hat:)))
- **Indermark:** (Guckt mich so an, als ob er bei mir ein drittes Auge entdeckt hätte:)) Das wäre schön...
- **Ich:** Oh, Mist, das ist Unfug. Das ist doch das SAT-Problem, (gucke den Unger an, er lächelt) da reduziert man doch alles drauf, es ist NP-vollständig.
- **Indermark:** Genau!
- **Indermark:** Wir haben ja einige Normalformen von Formeln behandelt...
- **Ich:** Ja, die Pränexnormalform, die Skolemnormalform und die Konjunktive Normalform.
- **Indermark:** Wie skolemisiert man denn eine Formel in Pränexnormalform, schreiben Sie mal einen Beispiel auf $\forall x \forall y \exists z \varphi'$. Wie skolemisiert man diese Formel? Erklären Sie bitte, nur grob.
- **Ich:** Habe das mit der Erweiterung der Signatur um ein zusätzliches Funktionssymbol und Ersetzung von $\exists z$ durch diese Funktion von den vorherigen Argumenten erläutert. In Details wollte er nicht gehen. Wollte also nur den Grundgedanken hören.
- **Indermark:** Diese Eliminierung von Existenzquantoren beruht auf einem Satz, wie heisst er?
- **Ich:** (keinen Schimmer) Ähh...
- **Indermark:** Das ist der berühmteste Satz der Informatik!
- **Ich:** Nee, kenne ich nicht, aber wir haben das auch nicht in der Vorlesung gemacht.
- **Indermark:** OK, ist nicht so schlimm. Das ist der Satz von Tarski. (Hab zum ersten Mal in meinem Leben davon gehört. Allerdings wurde ein Kumpel von mir am nächsten Tag von Indermark in Logikprogrammierung geprüft und er wurde auch nach diesem Satz gefragt - also ist es durchaus sinnvoll, zu wissen, wie dieser Satz heisst und was er aussagt.)
- **Indermark:** OK. Sie sprachen von der SLD-Resolution. Worin unterscheidet sie sich von der linearen Resolution?

- **Ich:** Blablabla - die Unterschiede bzw. die Gemeinsamkeiten aufgezeigt.
- **Indermark:** Was ist denn in der SLD-Resolution nicht möglich?
- **Ich:** Habe ihm die Zeichnung der linearen Resolution aufgezeichnet und erklärt, dass man in der SLD-Resolution nur mit den Programmklauseln resolvieren kann und nicht auch mit den den vorherigen Resolventen wie bei der linearen Resolution. Habe auch erklärt, warum das so ist.
- **Indermark:** Wie sehen die Klauseln nach der Resolution aus?
- **Ich:** Im besten Fall hat die Resolvente einen Literal weniger, wenn man mit einer Tatsachenklausel resolviert. In anderen Fällen, wenn man mit Regeln resolviert, bleibt die Anzahl der Literale gleich oder wird sogar größer. (Habe das noch etwas ausführlicher erklärt).
- **Indermark:** Aha.
- **Indermark:** Wir haben auch SLD-Baum gemacht, da gibt es mehrere Suchstrategien...
- **Ich:** Ja, die Tiefensuche und die Breitensuche. (Hab mich dann noch über die Vorteile / Nachteile von beiden ausgelassen, sowie gesagt, dass man die Tiefensuche verwendet und warum man das so tut.)
- **Indermark:** Dafür brauchen wir die Knoten im SLD-Baum. Was sind sie?
- **Ich:** Konfigurationen. (Habe dann erklärt, was eine Konfiguration ist, sie aufgeschrieben - $G', []$)
- **Indermark:** Die SLD-Resolution läuft ja auf Hornklauselmengen. Was sind denn die Hornklausel?
- **Ich:** Die Definition gesagt. Habe dann auch gesagt, dass sie eine echte Einschränkung der Prädikatenlogik sind, dass das aber für die Praxis nicht so relevant ist, und dass es dafür einen effizienten Erfüllbarkeitstest für die Hornformel gibt.
- **Indermark:** Wie effizient er?
- **Ich:** Also er ist im Durchschnitt effizienter als bei den normalen Formeln, allgeimein kann man aber keine Effizienz angeben, da es ja immer noch die unendlichen Berechnungen gibt.

- **Indermark:** Also so stimmt das nicht mit der Effizienz, aber wir kommen noch dazu später (keine Ahnung, warum das nicht stimmt, aber wir sind „später“ nicht dazugekommen:)))
- **Indermark:** Da gab es bei der Logikprogrammierung mehrere Semantiken: die Deklarative, die Prozedurale und die Fixpunktsemantik. Erklären Sie mir mal die hmm... na nehmen wir doch mal die Fixpunktsemantik.
- **Ich:** Also dafür muss ich zuerst die $trans_{\mathcal{P}}$-Funktion erklären.
- **Indermark:** (Leuchtet auf) Ja, tun Sie das.
- **Ich:** Blablabla... hab bei der Definition eine kleine Ungenauigkeit gemacht, habe geschrieben, dass $A', \neg B'_1, \ldots, \neg B'_k \in M$.
- **Indermark:** Das stimmt aber nicht ganz genau, so, wie Sie das geschrieben haben, könnten diese Primformeln in verschiedenen Klauseln des Logikprogramms stehen.
- **Ich:** ???... Aaaah, ja, klar, es soll natürlich so sein, dass A' von den $\neg B'_i$ abhängt!
- **Indermark:** Genau, sie sollten zusammen in einer Regel stehen.
- **Ich:** und es gibt bei $trans_{\mathcal{P}}$ einen minimalen Fixpunkt...
- **Indermark:** (Unterbricht) Und warum gibt es ihn?
- **Ich:** (Hab nicht so richtig Plan. Weil es im Skript steht?:))) Ähhh, diese $trans_{\mathcal{P}}$ hat ja drei Eigenschaften: Monotonie, Stetigkeit...
- **Indermark:** Na ja, also Monotonie hat ja damit gar nichts zu tun. Und Stetigkeit folgt aus der Monotonie (irgendsowas hat er gesagt, weiss aber nicht mehr genau, kann mich auch irren)
- **Ich:** Ja, und $\langle \mathfrak{P}(Prim); \subseteq \rangle$ ist ja ein vollständiger Verband. Und daraus folgt das.
- **Indermark:** Ja... (noch irgendwas erklärt, woran ich mich nicht mehr erinnere, hat aber dann nicht weiter nachgebohrt)
- **Ich:** Blablabla - Fixpunkt erklärt und die einzelne Schritte von $trans_{\mathcal{P}}$ auf $\{\emptyset\}$ erläutert. Dann habe ich ihm die Fixpunktsemantik aufgeschrieben und erläutert.

- **Indermark:** Dann wenden wir das mal auf einen Beispiel an. Schreiben Sie das Programm $\mathcal{P}$ für Addition von natürlichen Zahlen auf.
- **Ich:** $add(X, null, X) \quad add(X, succ(Y), succ(Z)) : -add(X, Y, Z)$
- **Indermark:** Was wird denn im ersten Schritt von $trans_{\mathcal{P}}(\emptyset)$ generiert?
- **Ich:** Die Grundinstanzen von $add(X, null, X)$ blablabla (hab das halt ausführlich erläutert)
- **Indermark:** Und im zweiten Schritt?
- **Ich:** Blablabla - die Regel angwandt, um weitere gültige Aussagen des zweiten Schritts abzuleiten.
- **Indermark:** Es gibt ja noch außer Prolog die Spracha Datalog, die zum Beispiel bei den Datenbanken angewandt wird. Kennen Sie diese?
- **Ich:** Ja, natürlich. Habe ihm dann Datalog erklärt (Breitensuche, Fehlen von Funktionssysymbolen $\Rightarrow$ endliche Anzahl von Termen $\Rightarrow$ Möglichkeit der Erkennung unendlicher Berechnungen durch sich wiederholende Terme)
- **Indermark:** Moment mal. Sie sprachen von den endlich vielen Termen. Wovon gibt es denn dann endlich viel? (oder so ähnlich)
- **Ich:** ??? Tut mir leid, aber ich verstehe nicht ganz, worauf Sie hinauswollen:)
- **Indermark:** Wovon gibt es endlich viel in den Berechnungen?
- **Ich:** Ah, von Literalen.
- **Indermark:** Aha. (Wollte anscheinend das Wort Literal hören)
- **Indermark:** Und Sie meinten, dass man unendliche Berechnungen durch sich wiederholende Konfigurationen erkennen kann. Heisst es denn, dass wenn sich ein Term wiederholt, die Berechnung notwendig unendlich ist?
- **Ich:** Hmm... (also wenn er schon so fragt:))) Nein.
- **Indermark:** Genau. (Hat es dann noch weiter erklärt, meinte aber auch, dass wir sowas in der Vorlesung nicht gemacht haben)

- **Indermark:** So, wir machen zwar Logikprogrammierung, aber um den Compilerbau kommen wir nicht herum.
- **Ich:** (Was!???) Na ja, ich hab die Vorlesung nicht gehört...
- **Indermark:** Das macht nichts. Wir haben auch in der Logikprogrammierung etwas davon gemacht. Und zwar haben wir die kontextfreien Grammatiken auf eine gewisse Weise in Prolog dargestellt...
- **Ich:** Sie meinen die Definite Clause Grammars.
- **Indermark:** Genau. Erläutern Sie mir die Idee.
- **Ich:** Welche soll ich erläutern? Die mit *append* oder die mit Differenzlisten?
- **Indermark:** Machen Sie zuerst die mit *append* und einem Argument.
- **Ich:** Hab angefangen, es irgendwie komisch aufzuschreiben, hab mich dann aber direkt selber gefasst und die Umwandlung in Prologklauseln $a(w) : -b(w_1), c(w_2), append(w_1, w_2, w)$ hingeschrieben.
- **Indermark:** Ja, das ist schon die Umsetzung in die Prologklauseln. Wann wird denn ein w von dem Prädikat a akzeptiert?
- **Ich:** Blablabla (Zusammenhang zwischen dem Nichtterminalsymbol A und a, Bedeutung der Formel erklärt)
- **Indermark:** So, nun ist es so, dass es ziemlich umständlich ist, weil die *append*-Funktion immer wieder aufgerufen wird. Wir hatten eine Verbesserung mit Differenzlisten und zwei Argumenten. Schreiben Sie auf $a(v, w)$. Wann wird dieser Prädikat erfüllt?
- **Ich:** Wenn $v - w$ aus dem Nichtterminal A ableitbar ist.
- **Indermark:** Was heisst denn konkret $v - w$?
- **Ich:** Ja, zum Beispiel, wenn $v = a_1a_2a_3$ und $w = a_3$, dann ist $v - w = a_1a_2$. Also w ist ein Suffix von v. (Das Wort „Suffix“ wollte er wohl hören)
- **Indermark:** OK, gehen Sie bitte kurz raus.

4 Fazit

Wie auch alle anderen Prüflinge vor mir fand ich die Atmosphäre in der Prüfung sehr locker und angenehm. Trotzdem bin ich während des gesamten Verlaufs recht nervös gewesen:)

Zu dem Stoff kann ich Folgendes sagen: Man muss nicht ALLES können, um eine 1.0 zu bekommen.Wie aus dem vorliegenden Protokoll leicht ersichtlich ist, habe ich nicht alles gekonnt und das hat meiner Note nicht geschadet. Man sollte die wichtigen Definitionen und Beweise können, sowie einen guten Überblick haben. Aber nicht irgendeine komische Lemma XY auf Seite Z, die mal in der Vorlesung aufgetaucht ist. Ich muss auch sagen, dass ich nach der Prüfung, als ich draußen warten musste, mit einer wesentlich schlechteren Note gerechnet habe, weil ich eben einige Sachen nicht wusste. Aber wie man sieht, ist es nicht schlimm, wenn man ein paar Nebensächlichkeiten nicht kann. Wie Professor Indermark mir bei der Besprechung sagte: „Sie können ja nicht alles wissen“:)))

Sollte jemand noch Fragen bezüglich dieses Protokolls haben, kann er mir gerne schreiben.