

Prüfungsprotokoll

Prüfer: Priv.-Doz. Dr. Thomas Noll und Priv.Doiz. Dr. Walter Unger

Fächer:

- Algorithmische Kryptographie (Unger)
- Programmanalyse & Compileroptimierung (Noll)
- Compilerbau (Noll)

Datum: 12.08.2008

Dauer: 45 Minuten

Note: 1.3

Vorwort

Gelernt haben wir in einer Dreiergruppe. Das ist auf jeden Fall zu empfehlen. Es gibt da zwar immer Motivationslöcher aber im Großen und Ganzen auf jeden Fall besser als alleine zu lernen.

Bis auf Krypto haben wir den Stoff komplett anhand von Skripten, Folien und Vorlesungsvideos gelernt, was natürlich zeitaufwendiger ist, da nur wenig Vorwissen vorhanden war.

Die Stimmung im Prüfungsraum war auf Seiten der Prüfer sehr gelassen. Ich war natürlich nervös, das hat sich aber recht schnell gelegt. Hab das für mich leichteste Themen an den Anfang gelegt. So hab ich etwas mehr Sicherheit bekommen.

Dieses Protokoll ist ein reines Gedächtnisprotokoll. Es kann sein, dass ich etwas vergessen hab oder die Reihenfolge etwas durcheinander gebracht hab. Desweiteren wurde dieses Protokoll in keinsten Weise durch die Prüfer abgesegnet.

Kryptographie

Ich wurde gefragt, was die Sicherheitsaspekte von RSA sind. Hab angefangen mit dem Aufbau von RSA und was für Eigenschaften die Primzahlen haben sollten. Dr. Unger fragte dann nach der Sicherheit des letzten Bits. Hab von dem Orakel erzählt, das das letzte Bit voraussagen kann. Wenn so ein Orakel existiert, kann man durch 'geschicktes' Teilen durch 2 auf den Plaintext schließen. Dr. Unger wollte das etwas genauer wissen, konnte ich ihm aber leider nicht mit dienen. Er fragte danach, ob man ein Schlüsselpaar für das Verschlüsseln von Texten und gleichzeitig zum Unterschreiben benutzen kann. Kann man, sollte man aber nicht. Den Text sollte man hashen. Das war das Stichwort für die nächste Frage: Wie hashed man? Merkes-Meta erklärt, nur vergessen zu erwähnen, dass die benutzte Funktion eine Kompressionsfunktion ist.

Das nächste Thema war das Geburtstagsprotokoll (das letzte). Hatte leider aus Nervosität den Anfang vergessen. Bin aber recht schnell drauf gekommen und hab das Protokoll aufgeschrieben, brauchte aber zwischendurch immer ein paar kleine Hilfen. Dr. Unger hat auch noch gefragt, warum bei dem Protokoll $\text{mod } p$ gerechnet wurde. Hab ihm erzählt, dass dadurch die Ordnung durcheinander gebracht wird.

Das dritte Thema waren Wahlen. Dr. Unger fragte mich nach dem Protokoll mit der ϕ -Funktion. Hab ihm das Protokoll von Anfang an bis zur Veröffentlichung der Stimme eines Wählers erklärt, auch wie die ϕ -Funktion arbeitet und dass der Wähler seine Stimme b suchen muss, damit sie zu seinem Wunschkandidaten passt.

Letztes Thema waren Zero-Knowledge-Proofs (sein Lieblingsthema). Ich sollte ihm das Protokoll

zur Kenntnis eines diskreten Logarithmus erklären. Hab ihm das Protokoll aufgeschrieben. Die nächste Frage war dann, ob die parallele Ausführung von ZKP-Protokollen auch ZKP ist. Eine grobe Erklärung des konstruierten Protokolls, bei dem es nicht funktioniert reichte ihm. Hab dann noch gesagt, dass die ZKP-Protokolle hintereinander ausgeführt werden können. Dazu wollte Dr. Unger wissen wieso. Ein kurzer Satz über den Simulator reichte ihm da aber auch schon.

Programmanalyse & Compileroptimierung

Die erste Frage war, was MOP und was MFP ist. Hab die beiden Verfahren erklärt. Die nächste Frage war, wie bei MFP so ein Gleichungssystem aussieht. Mit einem kleinen Beispiel und den zugehörigen Transferfunktionen war Dr. Noll schon zufrieden. Nun sollte ich erklären, warum es einen kleinsten Fixpunkt gibt. Hab was erzählt von Halbordnungen mit ACC-Eigenschaft und der Monotonie der Funktion auf der Halbordnung aus der die Stetigkeit folgt. Er war recht schnell zufrieden.

Als nächstes sollte ich was zu der Konstantenfaltung erzählen. Angefangen hab ich mit der zugehörigen Analyse, der Reaching-Definition, zuerst auf einem SLC-Programm, dann auf einem IC-Programm. Dr. Noll wollte daraufhin wissen, ob die RD-Analyse bei MOP und MFP immer die selben Ergebnisse liefert. Hab erklärt, dass MOP im Normalfall bessere Ergebnisse liefert. Daraufhin sollte ich das anhand eines Beispiels zeigen, dass MOP nicht gleich MFP ist. Graph gezeichnet und ein bisschen erklärt. Die letzte Frage war dann, wann MFP und MOP gleich sind. Kurz die Distributivität erklärt und dann war das Thema schon vorbei.

Compilerbau

Dr. Noll fing an mit einer Beispiel-Grammatik. Ich sollte ihm sagen, ob eine Top-Down-Analyse möglich ist. War es nicht, da es keine LL(0)-Grammatik war. Seine nächste Frage zielte auf einen möglichen Lookahead ab. Hab ihm erklärt, was ein Lookahead bewirkt und dass es hier auch erstmal nichts bringt ($k=1$ war zu kurz). Da ich eh wusste, was als nächstes kommen sollte, hab ich direkt gesagt, dass man die Grammatik durch Linksfaktorisierung 'reparieren' kann. Hab das dann durchgeführt und die la-Mengen berechnet. Dazu hat er dann auch direkt gefragt, wie ich die Mengen Schritt für Schritt berechnet hab.

Die nächste Frage war nach einer Bottom-Up-Berechnung. Ich sollte die LR(0)-Mengen zu der ursprünglichen Grammatik aufstellen. Bei der dritten Menge brach er ab mit der Frage, ob mir schon was auffällt. Es trat ein shift/reduce-Konflikt auf. Da wollte Dr. Noll genauer wissen, wieso das ein Konflikt ist. Hab genauer erklärt, wann geschifted wird und wann reduced wird. Seine nächste Frage war, wie man so einen Konflikt lösen kann. Angefangen hab ich mit LR(1), worüber er aber nichts wissen wollte, also hab ich was zu SLR(1) erzählt und wie die action-Function aufgebaut ist. Dr. Noll hat nicht danach gefragt, hab ihm trotzdem erzählt, wann SLR(1) einen s/r-Konflikt erzeugt.

Die letzte Frage war zu Code-Generation. Da ich das Kapitel nur angekratzt hatte und noch nicht mal die Frage verstanden hab, konnte ich dazu leider nichts sagen. Das hat mich (leider) auch die 1.0 gekostet.