Gedächtnisprotokoll

Prüfer: Prof. Hromkovic (H)
Beisitzer: Joachim Kupke (K)
Fächer:

- Algorithmische Kryptographie (nach Delfs/Knebl),
- Effiziente Algorithmen (WS 03/04),
- Compilerbau (leider nicht gehört - nach Skripten von s-inf.de gelernt..)

Dauer: ca. 35 min
Note: 1.0
Datum: 19.03.04

Hier ist so der grobe Ablauf der Prüfung, wobei ich mich leider nicht mehr an die genauen Zwischenfragen erinnern kann, da ich recht aufgeregt war.. Der Beisitzer hat bei Krypto und Compilerbau auch hin und wieder Zwischenfragen gestellt, da ich mich manchmal nicht so klar ausgedrückt hatte.. war nen bisschen unangenehm, da ich oft nicht so recht wusste, worauf er hinaus wollte. Ansonsten war die Atmosphäre aber sehr angenehm! Bei Effiziente hatte ich den Eindruck, dass Hromkovic etwas ungeduldig wurde - viele Sachen hab ich gar nicht zu Ende erläutert, er hat dann immer mit dem Kommentar abgebrochen: „ok, der Rest ist klar.. “

Krypto

H: Was ist denn der Unterschied zwischen symmetrischen und asymmetrischen Verfahren?
P: symmetrische Verfahren: ein Schlüssel; PK: zwei Schlüssel, basierend auf Einwegsfunktionen..
H: Was sind Einwegsfunktionen?
P: Einfach zu berechnen, schwierig zu invertieren
H: Welche gibt es?
P: Modulares potenzieren (RSA), DL (El Gamal), Modulare Wurzeln (Rabin)
H: Wie funktioniert RSA?
P: (Formel aufgeschrieben..)
H: Und was ist die Umkehrfunktion von RSA?
P: ?? (ich hatte keinen Plan, was er von mir wollte und hab mir einen zurecht gestammelt von wegen e-te Wurzel.. er wollte auf den Diskreten Logarithmus hinaus oder so..)
H: Für die Entschlüsselung: wie sieht denn der Chinesische Restsatz genau aus?
P: (Da wusste ich nur noch: Ist Isomorphismus mit $\Phi : \mathcal{Z}_n \rightarrow \mathcal{Z}_p \times \mathcal{Z}_q$. Er wollte aber die genaue Formel haben..)
H: Erläutern Sie El Gamal
P: Formel zum Ver- und Entschlüsseln aufgeschrieben
K:Wie wird primitive Wurzel g erzeugt
P: ?? (Mist, das wusste ich mal)
H: (hat dann irgendwas dazu erläutert, wobei ich aber zu nervös war, um das auch nachzuvollziehen..)

H: Verfahren nach Rabin?
P: Prinzip zum Ver- und Entschlüsseln aufgeschrieben
K:Wie werden denn die Wurzeln genau berechnet?
P: wieder mit dem Chinesischem Restsatz, aber denn konnte ich ja vorhin schon nicht..

H: Nachteil/Problem bei dem Verfahren?
P: ??
H: Wenn man dann mehrere Wurzeln hat..
P: Ach ja, welche Nachricht dann die Richtige ist, im Allgemeinen sind die ja nicht in natürlicher Sprache verfasst, deswegen wird das Jacobi-Symbol angehängt..
H: ok, das reicht... Fiat Shamir?
P: vereinfachtes Verfahren aufgeschrieben

Effiziente Algorithmen

H: Beginnen wir mit NP-Vollständigkeit: welche Reduktionen kennen Sie?
P: Hab dann alle aus der Vorlesung aufgezählt: NTM$\leq_p$Sat$\leq_p$3Sat$\leq_p$3-Col$\leq_p$3-Col planar$\leq_p$3-Col planar Grad 4, 3Sat$\leq_p$Clique$\leq_p$IS$\leq_p$VC (da hat er dann abgebrochen)
H: Reduktion von NTM auf Sat: erläutern sie die Variablen
P: $C(i,j,t)$, $S(k,t)$, $H(j,t)$ aufgeschrieben und erläutert
H: Kennen sie auch die einzelnen Bestandteile der Formel
P: Ja. A bis G erläutert
H: Wir hatten ein Verfahren, wo wir exponentielle Komplexitäten in Kauf genommen haben..
P: Ähh..
H: Das Kapitel hieß „Lowering worst case Complexity..“
P: Ach ja: D&C 3Sat, Komplexität $O(r1,84^n)$, Ansatz für Divide & Conquer Strategie erläutert,
H: Können sie die Rekurrenzgleichung aufgeschrieben?
P: Ja: $T(n,r) = 54r + ..$
H: ok, das reicht.. Was hat es denn mit den Diamanten auf sich?
P: Diamant aufgezeichnet, Prinzip ist.., die werden für Reduktion von HC und RHC und für den pathologischen Fall von TSP benötigt
H: Reduktion HC auf RHC?
P: Verfahren erläutert, wurde dann gegen Ende abgebrochen
H: Erläutern Sie Christofides
P: Algorithmus erläutert, ist 1,5-approximativ für Δ-TSP
H: Bei der Abschätzung: wie kommen die Matchings zustande (oder so ähnlich)
P: habe dann den Pfad für H_Opt aufgezeichnet und erläutert, dass die Knoten v_i die aus dem Matching sind.. das wollte er wohl nur hören

Compilerbau

H: Kommen wir zu Compilerbau: Syntaxanalyse?
P: Zerlegung der Symbolfolge, die der Scanner liefert, in syntaktische Einheiten, Fehlererkennung. Hilfsmittel: LL(k)/LR(k) Grammatiken
H: Können sie auch die Definitionen dafür aufschreiben?
P: Ja, hab Definition für LL(k) aufgeschrieben
H: Können Sie auch die Definition der $first_k$ Mengen?
P: Ja. (begann aufzuschreiben, kam aber irgendwie ins stocken, hab das dann verbal erläutern (sind terminale Anfänge einer Kette α der Länge k..))
K:Wie teste ich, dass eine Grammatik LL(k) ist?
P: Äh.. weiß ich jetzt nur für den Fall k=1: la-Menge der Alternativen sind disjunkt (Definition für la-Menge aufgeschrieben)
H: ok, warten Sie bitte draußen