Prüfungsprotokoll Theoretische Informatik

Datum:	08.04.02
Prüfer:	Hromkovic
Gebiete:	Effiziente Algorithmen (Algorithmics for Hard Problems, Hromkovic) Kryptographie (Vorlesung Unger) Compilerbau (Vorlesung Indermark)
Dauer:	40 min.
Note:	1.0

****
****

H: Reihenfolge?
P: EA, Krypto, CB!

Effiziente Algorithmen

H: Erzählen Sie mir etwas über das Prinzip Teile&Herrsche
P: Zerlegung der Probleminstanz in kleine Teile, danach rekursive Berechnung der Teilprobleme, am Ende Zusammensetzung der Lösung des Ursprungsproblems aus den Lösungen für die Teilprobleme
H: Kennen Sie ein Beispiel?
P: Beispiele sind Sortieralgorithmen, Gegenbeispiel Fib
H: Erklären Sie wie die Multiplikation großer Zahlen funktioniert
P: Erkläre, dass sich A aus A_1 und A_2 zusammensetzt und wie A_1 und A_2 genau aufgebaut sind (Kap. 2.3.4), das gleiche für B. Will die Formel für die verbesserte Version von A*B hinschreiben und gerate beim Teil $(A_1-A_2)(B_2-B_1)$ ins straucheln, weil ich ziemlich nervös war. H versuchte zu helfen, aber irgendwie sah ich den Wald vor Bäumen nicht.
H: egal, machen wir weiter. Erzählen Sie mir etwas über parametrisierte Komplexität.
P: Bei der PC teilt man den Lösungsraum in k möglicherweise unendliche Partitionen auf . Hierbei ist eine Parametrisierung folgendermaßen definiert: (Def. 3.3.1.1 runtergeleiert).
H: Kennen Sie ein Beispiel?
P: Ja, den Algorithmus ParVC. Er basiert auf zwei Beobachtungen (3.3.2.2 und 3.3.2.3 geschildert). Dann habe ich den Algorithmus (3.3.2.4) hingeschrieben und auch die Laufzeitabschätzung erklärt.
H: Gut, dass reicht soweit.

Kryptographie

H: Erklären Sie mal RSA
P: RSA und Verschlüsselung/Entschlüsselung erklärt.
H: Welche Eingaben sind für RSA zulässig?
P: für w muss gelten, dass w<n ist.
H: Zeigen Sie mir, dass die Entschlüsselung eindeutig ist
P: Es gibt da drei Fälle …(Alle drei Fälle erklärt, so wie sie im Skript standen). Für den dritten Fall darauf hingewiesen, dass es diesen Fall nicht gibt, da w<n ist.
H: Erklären Sie, was man mit Public Key-Algorithmen macht

P: Hab die Vorteile von Public-Key Algorithmen gegenüber früheren Ansätzen erklärt (D_A kann nicht aus E_A bestimmt werden, Oneway-Funktionen usw).
H: und wie sieht es da mit RSA aus?
P: Habe erwähnt, dass man RSA heutzutage für die Normale Verschlüsselung von Nachrichten, in Signaturen und in verschiedenen Protokollen verwendet
H: Welche Protokolle gibt es da?
(habe hier nicht gerafft, dass er nur generell eine Auflistung aller Protokolle haben wollte). So habe ich dann überlegt, in welchen Protokollen RSA eine Rolle spielt und angefangen sie aufzuzählen. Das ging ihm wohl zu langsam.
H: Erklären Sie mir Zero-Knowledge-Proof für 3-Färbbarkeit
P: Zunächst erklärt, was verschließbare Boxen sind und was Dreifärbbarkeit eines Graphen ist, dann das Protokoll erklärt.
H: ok, das reicht

Compilerbau

H: Erklären Sie die Phasen eines Compilers
P: Es gibt 2 Phasen: Analysephase und Synthesephase. In der Analysephase geht es allgemein darum, die syntaktische Struktur eines Programm zu erfassen und Fehlererkennung zu betreiben. Sie spaltet sich in 3 Phasen auf: 1. Lexikalische Analyse: Erkennung von Symbolen, Trennzeichen Kommentaren; Hilfsmittel endliche Automaten, Ergebnis ist eine Folge von Symbolen. 2. Syntaktische Analyse: Erkennung des hierarchischen Aufbaus des Programms; Hilfsmittel CFG, Ergebnis: Ableitungsbaum. 3. Semantische Analyse: Erkennung der statischen Semantik, Kontextabhängigkeiten, Typinformationen; Hilfsmittel: attributierte Grammatiken, Ergebnis: attributierter Ableitungsbaum
2. Phase: Synthesephase: Optionaler Zwischencode, Optimierungen, Unter effizienter Ausnutzung des Registersatzes der aktuell vorliegenden Prozessorarchitektur Generierung von Maschinencode.
H: Kommen wir zur syntaktischen Analyse. Was sind LL(k)-Grammatiken und was macht man damit?
P: Definition von LL(k) hingeschrieben und erklärt, dass man die in der Top-Down-Analyse zur deterministischen Simulation des nichtdeterministischen Top-Down-Analyseautomaten mit k-look-ahead auf der Eingabe verwendet.
H: ja richtig. Das kann man dann ja auch graphisch darstellen (malt einen Baum auf und zeichnet darin w und A ein (hat er alles über kopf gemacht, deshalb musste ich erst zweimal hingucken, was er da eigentlich aufgemalt hat) und erklärt, was er da aufgemalt hat). Können Sie so was auch für LR(k)-Grammatiken aufzeichnen?
P: Ähhh, *Grübel* (hab ne kurze Zeit anhand der Definition von LR(k) drüber nachgedacht, wie so was aussehen könnte)
H: Sie kennen nicht die Definition von LR(k)?
P: Doch, klar. Anhand dessen überlege ich ja gerade, wie so was aussehen könnte.
H: *grinst*. Dann schreiben Sie die mal auf.
P: schreibe die Definition von LR(k) hin
H: Ja, Danke. Warten Sie bitte einen kurzen Moment draußen.

P.S.: Die Prüfung von H war wirklich superfair. Auch hat er mir durch seine ruhige Art ziemlich schnell die anfänglich doch ziemlich starke Nervosität genommen (weshalb er mir wohl auch den hirnrissigen Fehler bei der Multiplikation großer Zahlen nicht übel genommen hat). Auch lässt er einen mal ne kurze Zeit überlegen und hilft, wenn man etwas falsch gemacht hat gerne weiter, damit man es doch noch hinbekommt. Desweiteren

kann man an den Kommentaren und der Mimik von H recht gut abschätzen, ob etwas falsch bzw. richtig ist, was man gerade erzählt. Viel Erfolg bei Eurer Prüfung!