Prüfungsprotokoll Theoretische Informatik

Prüfer:	Prof. Dr. K. Indermark, Dr. W. Unger
Fachkombination:	Algorithmische Kryptographie (Unger)
	Parallele Algorithmen (Unger)
	Compilerbau (Indermark)
Datum:	11/2005
Dauer:	45 Minuten
Vorbereitungszeit:	14 Wochen, davon 8 effektiv
Note:	1.0

Bemerkungen

Die Fragen in diesem Prüfungsprotokoll sind nur sinngemäß wiedergegeben. Abgesehen davon kann ich leider nicht für die Vollständigkeit des Protokolls bürgen, da ich aufgrund der Nervösität die ein oder andere Frage vergessen habe. Ich hoffe, dass das Dokument dennoch hilfreich ist.

Obwohl ich sehr nervös war, empfand ich die Atmosphäre in der Prüfung als angenehm und konzentriert. Die Reihenfolge der Fächer konnte ich selber bestimmen. Die Fragen wurden verständlich formuliert. Die Prüfung begann entgegen meiner Erwartung nicht einer Einleitungsfrage (z.B. „Welche Protokolle gibt es?“), stattdessen ging es direkt in die Tiefe. Walter lässt bei den Antworten einen größeren Spielraum und fragt gerne nach Ideen. Dafür hält er sich nicht an einfachen Zusammenhängen auf, so dass ich das ein oder andere mal mit „Glaub ich Dir, nächste Frage“ unterbrochen wurde. Prof. Indermark stellt ausführlichere Fragen, die oft durch eine Erklärung eingeleitet werden. Die Antworten sind hier genau auf den Punkt zu bringen und die Definitionen sollte man auf jeden Fall beherrschen.

Natürlich sollte jeder die Literatur, die ihm liegt, selber aussuchen, dennoch möchte ich an dieser Stelle ein paar Empfehlungen aussprechen.

- **Algorithmische Kryptographie**: Die Vorlesungsfolien decken zwar das gesamte Thema ab, enthalten jedoch einige Fehler. Es wurde jedoch eine Errata-Liste erstellt. Ich habe noch aus den Büchern [DK02] und [Wät04] gelernt.
- **Parallele Algorithmen:** Auch hier enthalten die Vorlesungsfolien leider viele Fehler und sind nicht immer ganz verständlich, da gewisse Abbildungen fehlen. Allerdings wird die Vorlesung fast vollständig (mit Ausnahme von dem Kapitel über Einbettungen) durch zwei sehr gute Bücher abgedeckt. Wer die Vorlesung wirklich verstehen will, sollte sich auf jeden Fall [GR90] und [HKP$^+$05] angucken!
- **Compilerbau** lässt sich ausgezeichnet mit dem Skript lernen. Besonders hilfreich waren für mich jedoch auch die *Vorlesungsvideos* aus dem SS 2005, da

Prof. Indermark in der Vorlesung viele Dingen „zwischen den Zeilen“ erwähnt hat, die im Skript nicht so deutlich hervorgehoben werden. Das sog. „Drachenbuch“ [ASU88] ist zwar eine gute Ergänzung, hat mir aber nicht wirklich bei der Vorbereitung geholfen.

Prüfungsfragen

Algorithmische Kryptographie (15 Min.)

- Zero Knowledge Proof
  - Wie lautet die Definition von Zero Knowledge Proof (ZKP)?
  - Welche ZKPs kennst Du?
  - Nachdem ich ein paar aufgelistet hatte. Kriegt Du den ZKP von der Gleichheit zweier diskreter Logarithmen hin?
  - Wie sieht der ZKP beim einfachen Fiat-Shamir (Identifikation) aus?
  - Beweise die ZKP-Eigenschaft vom einfachen Fiat-Shamir-Protokoll. (Simulator aufstellen)
  - Warum muss der Simulator überhaupt den Verifier aufrufen?
  - Kann man ZKP parallel ausführen? (im Allgemeinen nicht, Gegenbeweis durch Protokoll)
- Wahlprotokolle
  - Welche Wahlprotokolle kennst Du und wie lautet die jeweilige Idee?
  - Speziell beim Wählen durch Mischen (3. Protokoll): Warum können wir bei der ϕ-Funktion raten? Entspräche das nicht der Bestimmung des diskreten Logarithmus'?
- Wahlprotokolle
  - Welche Wahlprotokolle kennst Du und wie lautet die jeweilige Idee?
- Elektronisches Geld
  - Welche Verfahren kennst Du und wie lautet die jeweilige Idee? (Verfahren von Shimon-Even, danach das Online-System mit den Ideen: Unterschreiben einer leeren Nachricht = Münze, Transskripte umformen usw.)

Parallele Algorithmen (15 Min.)

- Cole-Algorithmus
  - Erzähl mir was zum Algorithmus von Cole!
  - Welche Knoten sind aktiv während eines Durchlaufs?
- Einbettungen

    * Wie bettet man einen CCC in einen HQ ein?
- Gossip
    * Wie ist das Problem definiert?
    * Wie schnell geht Gossip auf Clique im 2-Weg-Modus?
    * Wie schnell geht Gossip auf Clique im 1-Weg-Modus?
    * Mit welchen Ideen haben wir die untere Schranke im 1-Weg-Modus ermittelt? (1. Network Counting Problem, 2. Kommunikations-Matrix, 3. Verteilung der Werte in der letzten Runde, Gleichungssystem)
- Bonusfrage: Was ist Dein Lieblingsbeweis bei den beiden Fächern? (Reduktion von CVP auf DFS)

Compilerbau (15 Min.)

- Syntaxanalyse
    * Wie sind $LR(0)$-Mengen definiert?
    * Was bedeutet es, wenn $[A \rightarrow \alpha\cdot]$ und $[B \rightarrow \beta\cdot]$ in einer Auskunft auftaucht?
    * Wie sind follow-Mengen definiert?
    * Wann liegt ϵ in der follow-Menge? (Startsymbol)
    * Wie sind la-Mengen definiert?
    * Anhand der la-Mengen. Wann ist eine Grammatik $\in LL(1)$?
    * Wofür brauche ich bei der TD-Analyse die la-Mengen? (um die Ableitungsschritte deterministisch zu machen)
    * Was passiert bei der $SLR(1)$-Analyse?
- Semantische Analyse
    * Wovon hängt ein Attribut im Ableitungsbaum ab? (wenn synthetisch vom Unterbaum, wenn inhertit vom Oberbaum)
- Codegenerierung
    * Strike vs. nicht-strikte Semantik.
        · Wo ist der Unterscheid? (Verlagerung in Kontrolle, Sprungziel, erklärt wie die *nbt*-Funktion aufgebaut ist)
        · Wie sieht das denn konkret beim $IF...THEN...ELSE$-Befehl aus?
        · Wie funktioniert das nun bei NOT und AND?
        · Worin besteht denn nun bei AND genau der semantische Unterscheid? (im zweiten Kommando könnte eine Endlosrekursion aufgerufen werden, die bei der nicht-strikten Semantik übersprungen würde, falls der erste Ausdruck *false* ergäbe)

* Wird wird denn Code nun verarbeitet? (die Frage zielte auf den Stack-Code ab und unter anderem auf die Auswertung in Umgekehrter Polynischer Notation)
* Wie ist der Prozedurkeller aufgebaut?
* Was bedeutet der statische Verweis? Wie erreiche ich die sichtbare Umgebung?

Literatur

[ASU88] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilerbau - Teil 1.* Addison-Wesley, 1988. AHO a 88:1 1.Ex.

[DK02] H. Delfs and H. Knebl. *Introduction to Cryptography.* 2002.

[GR90] Alan Gibbons and Wojciech Rytter. *Efficient parallel algorithms.* Cambridge University Press, New York, NY, USA, 1990.

[HKP+05] J. Hromkovič, R. Klasing, A. Pelc, P. Ružička, and W. Unger. *Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance.* Springer Monograph. Springer-Verlag, 2005.

[Wät04] Dietmar Wätjen. *Kryptographie.* Spektrum Akademischer Verl., 2004.