Prüfungsprotokoll der mündlichen Prüfung Grundlagen der Theoretischen Informatik 25310 (Bachelor Informatik)

<u>Prüfer:</u> Prof. Dr. Verbeek
<u>Semester der Prüfung:</u> SS 2010
<u>Datum der Prüfung:</u> 25.11.2010
<u>Dauer:</u> ca. 40 min

Hier sind sicherlich nicht alle Fragen, jedoch in meinen Augen die wichtigsten.

Berechenbarkeit:
Wann ist eine Menge rekursiv (entscheidbar) ?
-> Charakteristische Funktion...
Wann ist eine Menge rekursiv aufzählbar ?
Was ist Menge P(1) ?
-> Die Menge der einstelligen partiell-rekursiven Funktionen
Nennen Sie ein Beispiel einer rekursiv-aufzählbaren Menge:
Nennen Sie eine Menge die rekursiv-aufzählbar aber nicht rekursiv ist ?
-> Kphi (Selbstanwendbarkeitsproblem und Halteproblem erklärt und definiert)
Was ist Uphi ?
-> Definition und Erklärung
Was ist Phi ?
-> Definition und Erklärung

Komplexitätstheorie:
Was ist NP ?
-> Informelle Erklärung und Definition
Wann ist ein Problem NP-Vollständig ?
-> NP-Hart und selber in NP
Wann ist Problem NP-Hart ?
-> Wenn alle Probleme in NP polynomiell reduzierbar auf dieses Problem ist
Genaue Erklärung und Definition
Was ist die Menge FP ?

Formale Sprachen:
Welche Sprachen werden von DEA erkannt ?
Was ist eine reguläre Sprache ?
Was ist eine kontextfreie Grammatik ?
Wie sieht die Regelmenge aus ?
Welche Abschlusseigenschaften haben

Zur Prüfung allg.:
Prof. Dr. Verbeek ist ein sehr netter Prüfer. Ihm geht es mehr darum, die Zusammenhänge verstanden zu haben anstatt wortgenaue Definitionen aufschreiben zu können. Ich hatte leider einen schlechten Tag und habe diese Prüfung nur knapp bestanden.

Prüfung Theoretische Informatik (Bachelor) 1657/1658

Prüfer:	Prof. Dr. Verbeek
Datum:	09.09.2008
Dauer:	25 Minuten
Note:	1.0

Berechenbarkeit

- **Was sind Wortfunktionen?**

  Erklärt. Funktionen der Form $f :\subseteq (\Sigma^*)^k \to \Sigma^*$. Dabei ist Σ^* eine Folge von Elementen aus dem Alphabet Σ.

  Hinweis auf Turingmaschine und wann eine solche Wortfunktion berechenbar ist. (Wenn TM M existiert mit $f = f_M$).

- **Gibt es etwas ähnliches für Zahlenfunktionen?**

  Berechenbarkeit der Zahlenfunktionen ($f :\subseteq \mathbb{N}^k \to \mathbb{N}$) definiert über Registermaschinen. Die Funktion f heißt berechenbar, wenn eine RM M existiert mit $f = f_M$.

- **Zusammenhang zwischen Zahlenfunktionen und Wortfunktionen?**

  Nummerierungen und insbesondere die Standardnummerierung ν_Σ.

- **Was ist eine Nummerierung allgemein?**

  Eine surjektive Abbildung.

- **Was kann man nun mit dem Wissen machen, das Zahlen- und Wortfunktionen zusammenhängen?**

  Hier die Rückführung der berechenbarkeit auf jeweils bekannte bb. Zahlen- bzw. Wortfunktionen erklärt.

  Also für $f :\subseteq \mathbb{N} \to \mathbb{N}$ und $g :\subseteq \Sigma^* \to \Sigma^*$ gilt (*ich wollte k-stellige Funktionen verwenden, Prof. Dr. Verbeek sagte, es genüge, es an einstelligen zu erklären*)

$$\begin{aligned} f \text{ bb } &\Leftrightarrow \nu_\Sigma f \nu_\Sigma^{-1} : \Sigma^* \to \Sigma^* \text{ bb} \\ g \text{ bb } &\Leftrightarrow \nu_\Sigma^{-1} g \nu_\Sigma : \mathbb{N} \to \mathbb{N} \text{ bb} \end{aligned}$$

- **Was ist φ? Ist φ bejektiv?**

  φ informell erklärt (Standardnummerierung von $P^{(1)}$). Ist nicht bijektiv, da ν_P nicht bijektiv ist (Erklärung, warum) und φ u. a. aus dieser Funktion zusammengesetzt ist.

- **Was ist die Folge davon, dass φ nicht bijektiv ist?**

  z. B. Äquivalenzproblem

- **Was ist das Äquivalenzproblem?**

  erklärt

- **Mit welcher Methode beweist man, dass Äq nicht r.a. ist?**

  Z.b. Diagonalisierung

- **Geht auch. Aber wie sieht es hier mit Reduktion aus?**

  Reduktion erklärt und damit auch den Ansatz, um damit das Äquivalenzproblem zu beweisen.

Komplexitätstheorie

- **Was ist das P-NP Problem?**

  Komplexitätsklassen aufgezeichnet (P, NP, zudem noch NP-vollständig). P-NP erklärt. Insbesondere warum es so interessant ist. P Probleme noch gut mit Rechnern lösbar, EXP nicht. *Wikipedia-Artikel zu P-NP Problem ist hier sehr interessant und zur Vorbereitung zu empfehlen. Insbesondere das Zeichnen der Klassen und der "Hierarchie´´.*

- **Was sind besondere Probleme in NP?**

  NP-vollständige Probleme

- **Warum sind diese so besonders?**

  Wenn eines mit polynomialem Aufwand lösbar wäre, dann sind alle NP Probleme auch in polynomialem Aufwand lösbar. $\leq_{pol}$ erklärt.

- **Wie sind NP-vollständige Probleme definiert?**

  M ist NP-vollständig, wenn NP-hart und $M \in NP$. Zudem Erklärung, was NP-hart bedeutet.

- **Was gibt es denn für NP-vollständige Probleme?**

  SAT, 3SAT, RUCKSACK, TSP, usw. Jeweils kurze Beschreibung.

- **Wie beweist man NP-hart und $\in NP$?**

  $\in NP$: Maschine (nd), die Lösung rät und polynomial Zeit benötigt (*siehe hier insbesondere den Lösungsansatz zu SAT/3SAT in der entsprechenden Kurseinheit*).

  NP-hart: Reduktion auf bekanntes NP-hartes Problem. Eines muss natürlich schon auf eine andere Weise bewiesen sein. Hinweis auf 3D-DOMINO aus Kurstext.

Formale Sprachen

- **Welche Sprachtypen kennen Sie (Chomsky)? Welche Automaten erkennen die Sprache?**

  Typ0 bis Typ3 aufgeschrieben (genaue Definition). Erklärt. Bei 2 und 3 KA und EA erklärt.

- **Werden in allen Fällen von den Maschinen im det. und ndet Fall die gleiche Sprache erkannt?**

  Nein. Erklärt (regulär ja, kf nein usw.).

Prof. Dr. Verbeek ist ein sehr freunlicher Prüfer. Die Prüfung hat mir persönlich, trotz einigem mulmigen Gefühls im Vorfeld, sehr viel Spaß gemacht. Das Thema ist meines Erachtens wirklich nicht leicht, aber trotzalledem machbar - auch eine gute Note kann mit etwas Aufwand erreicht werden. Ich glaube, es waren noch ein paar Fragen mehr. Sind mir aber nicht mehr eingefallen ... Lösungen ohne Gewähr!!!