Prüfungsprotokoll Vordiplom theoret. Informatik

Kurse 1653/1654

Prüfer: Prof. Dr. Weihrauch
Datum: 31.08.2007
Note: 1,0

- Wie ist Berechenbarkeit(bb)definiert?
- Kennen Sie eine nicht bb. Funktion? Warum nicht bb.?
- Wie ist es im N^k? Cantorische Tupelfunktion, den Berechenbarkeitsbegriff mit g=f pi durfte ich nicht bringen.
- Zusammenhang:Wort-Zahlenbb.?
- Bb. bei R. Roh-Darstellung, SigmaRoh, Auch hier durfte ich nicht erklären, warum die Multipl. von 3*0,3-Periode nicht funktioniert.
- Eigenschaften (U) u. (S), Zusammenhang mit utm und smn-Theorem.
- Er meinte, dass man bei smn nicht so recht den Zusammenhang zu (S) sehen könnte. Ich erklärte ihm dann gleich den Satz über effektive Komposition.
- Äquvalenzsatz von Roger (man spricht es mit deutschem „g“)
- Kennen Sie Probleme aus NP? RUCHSACK, HAMILTON, 3SAT.
- Erklären Sie RUCKSACK!
- Können Sie das P-NP-Problem erklären?
- ….

Er stellte mir 3-4 Fragen, die glaube ich an der Grenze zum Kursstoff sind und deren Antwort er nicht wirklich erwartete:

- Was ist das Besondere am utm-Theorem, die Phi_i`s sind doch sowieso bb, dann muss es doch u_{Phi} erst recht sein, oder? Er gab die Antwort: Weil man Nummerierungen von $P^{(1)}$ wählen könnte, so dass u_{Phi} nicht bb. ist. (Aber die heißen doch dann nicht Phi_i ???)
- Warum muss es bei der Typ-2-Maschine ein Einweg-Ausgabeband sein? Hier wusste ich die Lösung auch nicht, aber ich konnte ihm zeigen, dass ich seine Antwort verstehen konnte: Damit wird gewährleistet, dass es zu irgendeinem Zeitpunkt eine Ausgabe gibt. Sonst könnte die Maschine ihre Ausgabe immer wieder korrigieren, wäre auch in der Lage die Multiplikation zu berechnen (erläuterte ich), man hätte aber nie eine festgelegte Ausgabe der Maschine. Und sie gilt ja als reales Modell.
- Warum kann man RUCKSACK nicht in kürzerer Zeit lösen. Weil es kein systematisches Vorgehen gibt um die Gegenstände auszuwählen. Man kann also nur wahllos ausprobieren.
- Er sagte man würde ja alle Teilmengen ausprobieren. Wie viele Möglichkeiten gibt es da? Das ist ja gerade die Potenzmenge, also 2^n.

Meinen Vorgängern muss ich zu allen positiven Äußerungen über Prof. Weihrauch zustimmen. Haltet Euch an die Prüfungsberichte, versucht auch zu verstehen, unterhaltet Euch mit Kommilitonen (z.B. telefonisch) über den Stoff und übt Prüfung.
Überhaupt bin ich der Meinung, dass man alle Veranstaltungen der Uni nutzen sollte um Kommilitonen kennen zu lernen und sich ein Netzwerk aufzubauen. Ich glaube nicht, dass man das Studium ohne dieses schaffen kann. Viel Erfolg.

Gedächtnisprotokoll
Vordiplom-Prüfung Theoretische Informatik

Kurse: Einführung in die Theoretische Informatik A (01653)
Einführung in die Theoretische Informatik B (01654)

Prüfer: Prof. Dr. Weihrauch

Termin: 03.08.2007

Dauer: 20 – 25 min

Note: 1,0

Vorbemerkung: Zunächst begann Prof. Weihrauch mit einigen Fragen zu meinem Studium, um die Nervosität etwas abzubauen. Auch bei mir begann die Prüfung mit der Frage:

1. Wann nennen wir eine Zahlenfunktion berechenbar?

- Definition über Registermaschinen.
- Kurze Erläuterung der Registermaschinen.

2. Ist diese Definition sinnvoll?

- Einfachheit der Registermaschinen angesprochen.
- Weitere Definitionen der berechenbaren Zahlenfunktionen durch WHILE-Programme und μ-rekursive Funktionen skizziert.
- Register-berechenbar ⇔ WHILE-berechenbar ⇔ „μ-Berechenbar".

- Bisher nur Zahlenfunktionen betrachtet, also: Betrachtung von Wortfunktionen als etwas näher am gewöhnlichen Computer liegende Alternative.
- Definition der berechenbaren Wortfunktionen mittels Turingmaschinen bzw. Bandmaschinen; Äquivalenz beider Modelle.
- Mit der Idee der Nummerierung der Worte zeigt man die Äquivalenz von Turing-Berechenbarkeit und Register-Berechenbarkeit.

- Alle Modelle haben dieselbe Berechenbarkeitsstärke und deshalb kann man sagen, dass die Definition sinnvoll ist.
- Churchsche These und ihre Bedeutung

3. Gut, dann sind wir mit dem Thema fertig. Wann nennen wir eine Menge A rekursiv?

- Definition: Charakteristische Funktion cf_A von A berechenbar.
- Definition: $A = f^{-1}\{0\}$ für $f \in R^{(k)}$

- A rekursiv $\Leftrightarrow \mathbb{N}^k \setminus A$ r.a. $\wedge$ A r.a.

4. Wann heißt eine Menge rekursiv aufzählbar?

- Definition: $\exists f \in P^{(k)}$ mit $A = Def(f)$.
- Projektionssatz

5. Ist die Menge A der geraden Zahlen rekursiv?

- Ja. Beweis: Charakterisierung der (unendlichen) rekursiven Mengen durch wachsende, rekursive Funktionen (Satz 9.1.6).

6. Ist A = Bild(f) mit $f \in R^{(1)}$, f wachsend, auch rekursiv aufzählbar?

- Ja, dies folgt aus der Charakterisierung rekursiv aufzählbarer Mengen durchs Bild rekursiver Funktionen.

7. Geben Sie ein Beispiel für eine Menge an, die rekursiv aufzählbar, aber nicht rekursiv ist!

- K_φ (Selbstanwendbarkeitsproblem) und K_φ^0 (Halteproblem) definiert, dabei kurz erwähnt, was φ ist.
- Für K_φ kurz begründet, warum nur rekursiv aufzählbar.

8. Wir hatten da wichtige Sätze über Programmiersprachen… Wie lauten die?

- Zunächst die Definition von $\varphi : \mathbb{N} \to P^{(1)}$ angegeben, Erläuterung der Konstruktion durch Skizzierung der Schritte von BP nach BM nach $P^{(1)}$.
- Die Eigenschaften U und S von guten Programmiersprachen.
- utm-Theorem genannt und erläutert.
- smn-Theorem genannt und erläutert.

9. Aus utm- und smn-Theorem folgen die Eigenschaften U und S für φ nicht unmittelbar, oder?

- Nein, nur U. Man kann aber zeigen, dass für Nummerierungen von $P^{(1)}$, die das utm-Theorem erfüllen, gilt: Das smn-Theorem gilt genau dann, wenn P gilt. Dies ist der Satz über die effektive Komposition 8.4.4.

10. Kennen Sie einen weiteren wichtigen Satz zu Programmiersprachen?

- Äquivalenzsatz von Rogers genannt.
- Diesen und seine Bedeutung vor allem im Hinblick auf die Modellprogrammiersprache φ erläutert: Diese ist äquivalent zu jeder anderen Programmiersprache, die auch das utm- und smn-Theorem erfüllt. Somit stellt die etwas willkürliche Definition der Modellprogrammiersprache φ kein Problem dar.

11. Es wurde auch ein kleiner Ausflug in die Welt der Logik gemacht. Was können Sie mir dazu sagen?

- Den Begriff des Beweissystems erläutert.
- Man kann entscheiden, ob ein Wort w ein Beweis für einen Satz v ist (Menge U_X).
- Da die Menge U_X rekursiv ist, folgt aus dem Projektionssatz, dass die Menge T_X der im Beweissystem X ableitbaren Sätze rekursiv aufzählbar ist.
- Wir verfahren ähnlich wie bei der Standardnummerierung φ und ordnen jeder natürlichen Zahl i eine r.a. Wortmenge, die Menge der ableitbaren Sätze im Beweissystem, zu.
- Betrachtung von Wörtern „$1^n \notin K_\varphi$“ als Sätze; Definition einer Semantik für diese Wörter; Definition der Begriffe des korrekten bzw. des vollständigen Beweissystems.
- Die zwei Aussagen der Version des Gödelschen Unvollständigkeitssatzes und deren Bedeutung.

Fazit: Prof. Weihrauch kann ich als Prüfer nur empfehlen. Mir hat gefallen, dass ich teilweise recht ausführlich antworten konnte. Dies hat mir vor allem Anfangs sehr dabei geholfen, Nervosität abzubauen. Sehr gut fand ich auch, dass Prof. Weihrauch durch Überleitungen und Kommentare versucht, das starre Frage-Antwort-Spiel etwas aufzulockern.

In meiner Prüfung wurde Teil B gar nicht angesprochen – was vielleicht daran lag, dass ich ein paar Tage zuvor noch die Klausur zu diesem Kurs geschrieben habe. Ich glaube aber eher, dass es daran gelegen hat, dass wir mit Teil A schon genug Zeit verbracht hatten. In den Prüfungsprotokollen (welche bei der Vorbereitung übrigens sehr hilfreich sind) kann man ja schon sehen, dass Prof. Weihrauch den Schwerpunkt gerne auf Teil A legt.

Viel Erfolg bei Deiner Prüfung!

Vordiplomprüfung Theoretische Informatik A und B

Gedächtnisprotokoll

Prüfer: Prof. Dr. Weihrauch
Beisitzer: Prof. Dr. Heinemann
Datum: 30.07.2007
Dauer: ca. 30min

Kurze Einleitung durch Prof. Weihrauch mit Einordnung der beiden Kurse und kurzer Zusammenfassung der Inhaltsschwerpunkte.

Wann heißt eine Zahlenfunktion berechenbar?
Wann heißt eine Funktion μ–rekursiv?
Wie wurde die Berechenbarkeit von Wortfunktionen eingeführt?
Turingmaschinen und Bandmaschinen.
Wie lautet die Beweisidee zum Nachweis der Äquivalenz von Turingmaschine und Bandmaschine?
Was muss man bei der Bandmaschine machen, wenn bei der Turingmaschine den Kopf um ein Feld nach rechts bewegen will?
Welche Zeitkomplexität ist bei der Bandmaschine zu berücksichtigen?
Wie kann man den Zusammenhang zwischen berechenbaren Zahlen- und berechenbaren Wortfunktionen darstellen?
Wann ist eine Menge $A \subseteq N$ rekursiv?
Was bedeutet $f^{-1}\{0\}$? Hier wünschte sich Prof. Weihrauch die Darstellung: $f^{-1}\{0\} = \{ a \mid f(a) = 0 \}$
Wann ist eine Menge $A \subseteq N$ rekursiv-aufzählbar?
Wenn zwei Mengen A und B rekursiv sind, ist dann auch die Vereinigung rekursiv? Wie lautet die Beweisidee?
Wie lautet der Zusammenhang zwischen rekursiv und rekursiv-aufzählbar?
Gilt dies auch für die Umkehrung?
Wenn eine Menge rekursiv-aufzählbar ist und ihr Komplement ist es auch. Wie zeigt man, dass diese Menge auch rekursiv ist?
Kennen Sie eine Menge, die rekursiv-aufzählbar aber nicht rekursiv ist?
Wie lautet die Beweisidee hierzu?
Wie sieht das mit der Berechenbarkeit von reellen Zahlen aus?
Welche Art von Maschine wurde für die Darstellung der Berechenbarkeit eingeführt?
Was ist das Problem bei der Dezimaldarstellung und wie kann man es veranschaulichen?
Wie kann man das Problem beheben?
Welches Maschinenmodell wurde im Zusammenhang mit der nichtdeterministischen Komplexität eingeführt?
Was ist unter P und NP zu verstehen?
Welche NP-vollständigen Probleme kennen Sie?
Worum geht es beim Rucksackproblem?
Was bedeutet $A \leq_{pol} B$?
Was sind reguläre Mengen?
Wie lautet die Definition regulärer Mengen als Abschluss?
Kennen Sie ein Beispiel für eine Sprache, die nicht regulär, aber kontextfrei ist?
Warum ist diese Sprache nicht regulär?
Wie sehen die Ersetzungsregeln bei den kontextfreien Sprachen aus?
Ist die Vereinigung von regulären Mengen auch wieder regulär?
Wie sieht das bei Durchschnitt und Komplementbildung aus?
Wie lauten die anschaulichen Beweisideen hierzu?

Fazit: Den bisherigen Prüfungsprotokollen kann ich mich nur anschließen. Herr Prof. Weihrauch ist ein netter und fairer Prüfer. Die Atmosphäre war sehr angenehm, so dass sich die Nervosität sehr schnell legte. Allerdings war auch wenig Zeit für Nervosität. Nach der kurzen Einführung bzw. Überleitung zu den einzelnen Prüfungsteilen kamen die einzelnen Fragen sehr flüssig hintereinander. Die Formulierungen hier daher nicht zu wörtlich nehmen. Die einzelnen Formeln wünscht sich Prof. Weihrauch schon sehr genau. Bei Unsicherheiten kann er auch schon sehr bohrend nachfragen, um zu sehen, ob der Zusammenhang verstanden wurde. Die Protokolle waren zur Vorbereitung eine gute Grundlage, es kommen aber auch Formulierungen vor, die recht spontan wirken und so noch nicht protokolliert sind.
Aufgrund der Erfahrung kann ich Prof. Weihrauch uneingeschränkt als Prüfer empfehlen.
Viel Erfolg bei Eurer Prüfung!

Gedächtnisprotokoll

Vordiplom - Prüfung

1653 Theoretische Informatik A
1654 Theoretische Informatik B

Prüfer: Prof. Dr. Weihrauch
Termin: 19.02.2007
Dauer: 25 – 30 min
Note: 1,0

Teil A

1. Wann heißt eine Zahlenfunktion berechenbar?
Eine Zahlenfunktion heißt berechenbar, gdw. es eine Registermaschine gibt, die diese Funktion berechnet.

2. Welche Operationen sind in den Flussdiagrammen von Registermaschinen zugelassen?
addiere 1 in Register R_i; subtrahiere 1 in Register R_i; teste, ob in Register R_i eine Null steht

3. Wann heißt eine Wortfunktion berechenbar?
Eine Wortfunktion heißt berechenbar, gdw. es eine Turingmaschine oder eine Bandmaschine gibt, die diese Funktion berechnet.

4. Da haben wir also die Äquivalenz von Turing – Berechenbarkeit und Band – Berechenbarkeit. Wie ist die Beweisidee dazu?
Um zu einer Turingmaschine mit mehreren Bändern eine Bandmaschine mit nur einem Band zu konstruieren, klebt man die endlich vielen Bänder der Turingmaschine zu einem einzigen Band mit vielen Spuren zusammen. Man markiert in jeder Spur die Stellung des Kopfes durch überstreichen. Dazu erweitert man das Bandalphabet um die Menge der überstrichenen Symbole. Jede Spalte des neuen Bandes ist ein Tupel, dass man als <u>ein</u> Symbol des neuen Bandalphabetes betrachtet.

5. Was muss man bei der Bandmaschine machen, wenn man bei der Turingmaschine auf Band 3 den Kopf ein Feld nach links verschieben möchte?
Man muss auf Spur 3 der Bandmaschine das markierte Symbol suchen und dann die Markierung um ein Feld nach links verschieben.

6. Welche Zeitkomplexität hat die Bandmaschine im Vergleich zur Turingmaschine?
Der Zeitbedarf der Bandmaschine gegenüber der Turingmaschine ist quadratisch. Denn bei der Simulation eines Schrittes der Turingmaschine muss die Bandmaschine zum Suchen des markierten Symbols schlimmstenfalls einmal das gesamte Band hin und her durchlaufen.

7. Wann ist eine Menge A Teilmenge von N rekursiv?
Eine Menge A Teilmenge von N ist rekursiv, gdw. die charakteristische Funktion der Menge A berechenbar ist.
Eine Menge A Teilmenge von N ist rekursiv, gdw. es eine totale berechenbare Funktion f gibt mit $A = f^{-1}\{0\}$.

8. Was bedeutet $f^{-1}\{0\}$?
$f^{-1}\{0\} = \{a \mid f(a) \text{ elem } \{0\}\}$

9. Ist die Menge der Zehnerpotenzen 10^x rekursiv?

Ja, denn man kann eine Registermaschine angeben, die die charakteristische Funktion der Menge der Zehnerpotenzen berechnet (Aufzeichnen des Flussdiagramms war nicht notwendig).

10. Wann ist eine Menge A Teilmenge von N rekursiv - aufzählbar?

Eine Menge A Teilmenge von N ist rekursiv – aufzählbar, gdw. es eine partielle berechenbare Funktion f gibt mit A = Def (f).

11. Wenn eine Menge A rekursiv ist, ist sie dann auch rekursiv – aufzählbar? -> Ja.

Beweisidee?

Sei A rekursiv. Dann ist die charakteristische Funktion von A berechenbar. Man kann eine verallgemeinerte Registermaschine M angeben, die hält, wenn die charakteristische Funktion gleich 1 ist, wenn also x elem A ist, und sonst in eine Endlosschleife läuft. Die von der Registermaschine berechnete Funktion f_M ist berechenbar. Da A der Definitionsbereich von f_M ist, ist A rekursiv – aufzählbar.

12. Wenn eine Menge A rekursiv– aufzählbar ist, ist sie dann auch rekursiv?

Nur wenn das Komplement der Menge A auch rekursiv– aufzählbar ist, sonst nicht.

13. Kennen Sie eine Menge, die rekursiv– aufzählbar, aber nicht rekursiv ist?

K_{phi} = {i elem N | phi $_i$ (i) existiert}

14. Beweisen Sie, dass K_{phi} rekursiv– aufzählbar, aber nicht rekursiv ist.

Es sei eine partielle berechenbare Funktion h definiert durch h(i) := u_{phi} (i,i) = phi $_i$ (i) für alle i elem N. Nach dem utm – Theorem ist h berechenbar und es gilt K_{phi} = Def (h). Also ist K_{phi} rekursiv – aufzählbar.

Um zu zeigen, dass K_{phi} nicht rekursiv ist, genügt es zu zeigen, dass das Komplement von K_{phi} nicht rekursiv– aufzählbar ist. Es sei i eine beliebige natürliche Zahl. Dann gilt nach der Definition von K_{phi} : i elem N\ K_{phi} ⇔ i ~~elem~~ K_{phi} ⇔ i ~~elem~~ Def (phi $_i$). Daraus folgt N \ K_{phi} < > Def (phi $_i$). Es gilt damit N \ K_{phi} < > Def (phi $_0$), N \ K_{phi} < > Def (phi $_1$), usw. Da die Folge (Def (phi $_0$), Def (phi $_1$), …) alle rekursiv – aufzählbaren Teilmengen von N enthält, kann N \ K_{phi} nicht rekursiv – aufzählbar sein. Damit ist K_{phi} nicht rekursiv.

15. Wo findet man rekursiv – aufzählbare Mengen in der Logik?

In der Logik gibt es Beweissysteme. Ein Beweissystem besteht aus einer Menge von Axiomen und einer Menge von Schlussregeln mit deren Hilfe man aus den Axiomen immer weitere Sätze ableiten kann. Eine Eigenschaft dieser Beweissysteme ist, dass man entscheiden kann, ob ein Wort w einen Satz v beweist oder nicht. Für ein Beweissystem X wird also gefordert, dass die Menge Ux := {(v,w) | w beweist den Satz v} rekursiv ist. Nach dem Projektionssatz ist dann die Menge Vx := {v | (es gibt ein w) (v,w) elem Ux} rekursiv – aufzählbar. Damit ist die Menge Vx der im Beweissystem X beweisbaren, ableitbaren Sätze rekursiv – aufzählbar.

16. Kennen Sie den ersten Gödelschen Unvollständigkeitssatz?

Der erste Gödelsche Unvollständigkeitssatz besagt, dass jedes korrekte Beweissystem unvollständig ist. D.h. zu jedem korrekten Beweissystem gibt es einen wahren Satz, der nicht im Beweissystem bewiesen wird.

17. Wir hatten Nummerierungen und Notationen zur Verschlüsselung der Elemente abzählbarer Mengen. Was ist das Problem bei den reellen Zahlen?

Die reellen Zahlen sind nicht abzählbar und haben deshalb keine Nummerierung. Statt dessen

werden die reellen Zahlen durch unendliche Folgen von Symbolen über einem Alphabet verschlüsselt.

18. **Ja, es gibt z.B. die Dezimaldarstellung der reellen Zahlen, wobei die reellen Zahlen durch unendliche Dezimalbrüche dargestellt werden. Was ist das Problem dabei und wie lautet die Beweisidee dazu?**
    Bei der Dezimaldarstellung ist die Multiplikation von reellen Zahlen mit 3 nicht berechenbar. Um das zu zeigen, nimmt man an, dass eine Typ – 2 – Maschine unendliche Dezimalbrüche mit 3 multipliziert. Dann muss die Maschine insbesondere für Eingabe 0.333… korrekt arbeiten. Man nimmt z.B. an, dass bei dieser Eingabe das Ergebnis 0.999… ist. Wenn man die Berechnung der Maschine Schritt für Schritt verfolgt, gibt es einen Zeitpunkt t nach dem sie 0. auf das Ausgabeband geschrieben hat. Nach t Schritten hat die Maschine nicht mehr als die ersten t Symbole vom Eingabeband gelesen. Wenn man nun eine Zahl $p = 0.3^t400\ldots$ (also genau t Dreien hintereinander) in die Maschine eingibt, hat sie nach t Schritten „0.“ Auf das Ausgabeband geschrieben. Es ist aber $3 * p > 1$, die Ausgabe müsste also mit „1.“ beginnen. Da die Maschine auf dem Ausgabeband nicht nach links laufen darf, arbeitet sie für die Eingabe p nicht korrekt.

19. **Welche Darstellung verwendet man, um das Problem zu beheben?**
    Man verwendet die Cauchy – Darstellung der reellen Zahlen, bei der die reellen Zahlen durch schnell konvergierende Folgen rationaler Zahlen dargestellt werden (formale Definition der Cauchy – Darstellung war nicht erforderlich).

<u>Teil B</u>

20. **Wann ist eine Menge regulär?**
    Eine Menge ist regulär, gdw. sie von einer rechtlinearen Grammatik erzeugt wird oder gdw. sie von einem endlichen Automaten erkannt wird.

21. **Kennen Sie auch die Definition als Abschluss?**
    Eine Menge ist regulär, gdw. sie sich in endlich vielen Schritten aus der leeren Menge und den ein – elementigen Wortmengen durch wiederholtes Anwenden der Operationen Vereinigung, Komplexprodukt und Sternoperation erzeugen lässt.

22. **Nennen Sie eine Sprache die nicht regulär, aber kontextfrei ist und erklären Sie anschaulich, warum diese Sprache nicht regulär ist.**
    $L = \{a^n b^n \mid n \text{ elem } N\}$

23. **Prof. Weihrauch erklärte, wann ein Wort von einer determinierten Turingmaschine erkannt wird. Frage: Wann wird ein Wort von einer nichtdeterminierten Turingmaschine erkannt?**
    Ein Wort x wird von einer nichtdeterminierten Turingmaschine erkannt, wenn sie bei der Eingabe von x irgendeine Möglichkeit hat, mit der Ausgabe epsilon zu halten. D.h. wenn es irgendeinen Berechnungspfad gibt, der mit der Ausgabe epsilon endet.

<u>Fazit:</u> Herr Prof. Weihrauch ist ein sehr netter Prüfer, der durch seine ruhige und freundliche Art dazu beiträgt, dass sich die Prüfungsnervosität schnell verringert. Man hat genügend Zeit, über die gestellten Fragen nachzudenken und die Antworten zu formulieren. Die Prüfungsprotokolle sind sehr gut zur Vorbereitung geeignet. Dennoch wurden mir zwei Fragen gestellt, die bisher in den Protokollen nicht aufgetaucht sind (Nr. 2 und 8). Also bitte auf alles gefasst sein und viel Glück bei Eurer Prüfung!

Vordiplomprüfung Theoretische Informatik
Grundlagen der Theoretischen Informatik A und B (1653 und 1654)

Gedächtnisprotokoll

Prüfer: Herr Prof. Dr. Weihrauch
Beisitzer: Herr Prof. Dr. Heinemann
Datum: 13.07.2006
Dauer: ca. 30 min
Note: 1,3

1. Wann heißt eine Zahlenfunktion berechenbar?

2. Wie ist das denn mit den verallgemeinerten Registermaschinen?
   Siehe Skript – die Funktion f_M der verallgemeinerten Registermaschine ist nur berechenbar, wenn alle in ihr vorkommenden Funktionen berechenbar sind. Test analog.

3. Wann ist eine Menge Teilmenge N rekursiv?

4. Wann ist eine Menge rekursiv-aufzählbar?

5. Wie ist die Standardnummerierung phi definiert?

6. Wo sind denn eigentlich die Programme in der Standardnummerierung?

7. Wie sieht denn das utm-Theorem aus?

8. Wie sieht das smn-Theorem aus?

9. Was sagen die U- und die S-Eigenschaft?

10. Wie sieht der Satz von der effektiven Komposition aus?

11. Was ist denn eine Nummerierung?
    Eine surjektive möglicherweise partielle Funktion von N nach Menge M.

12. Kennen Sie denn eine Nummerierung?
    Ich habe mich auf die Schnelle für die Nummerierung der reellen Zahlen entschieden.

13. Wann heißt denn eine Menge ny-rekursiv?
    Definition 10.3.

13. Wenn zwei Mengen rekursiv-aufzählbar sind, wie sieht es dann mit der Schnittmenge aus?
    Auch die Schnittmenge ist rekursiv-aufzählbar.

14. Wie kann ich das beweisen?
    Laufenlassen von zwei Maschinen, die eine partielle Funktion berechnen, deren Definitionsbereich jeweils eine der Mengen ist. Die Elemente, für die beide Maschinen halten, sind in der Schnittmenge enthalten.

15. Was sind denn die rechtslinearen Sprachen, wie sieht die Grammatik dazu aus?

16. Wie sieht die Normalform dazu aus, wie kann ich eine Grammatik in die Normalform überführen?

17. Wie hängen den Zeit- und Bandkomplexität zusammen?
    Hier sind die beiden Ungleichungen t_M <= . . . und s_M <= . . . gefragt sowie eine kurze informelle Erläuterung dazu, wie es zu diesen Abschätzungen kommt.

14. Worum geht es bei GAP?

Herr Prof. Weihrauch ist ein sehr angenehmer Prüfer, die Prüfungsatmosphäre ist ruhig und wohlwollend. Jede Frage leitet er mit einigen Worten ein, das gibt dem Prüfling Zeit sich auf das entsprechende Thema einzustellen. Er bringt auch selber noch ein paar anschauliche Erklärungen, da kann einem selbst in der Prüfung noch so manches Licht aufgehen. Die zentralen Definitionen und Sätze müßen einfach sitzen, Blatt und Stift liegen bereit um sie aufschreiben zu können. Anschließend sollte man auch noch in der Lage sein, ein paar Erklärungen zu bringen, was mit den Formalismen eigentlich gemeint ist.

Ich schließe mich der Meinung eines anderen Prüflings an – Herr Prof. Weihrauch weiß sehr schnell, ob der Prüfling sich ernsthaft mit der Materie auseinandergesetzt hat. Es kommt ihm auch besonders auf ein Verständnis dessen, was man als Prüfling erzählt, an und nicht nur darauf, daß Definitionen wiedergegeben werden.

Wie Ihr seht wurde bei mir wirklich die gesamte Stoffbreite zumindest kurz angesprochen. Auf eine ausführlich Beantwortung der Fragen hier im Protokoll verzichte ich, daß haben einige Kommilitonen in ihren Protokollen schon ganz ausgezeichnet getan!

Insgesamt sind die Prüfungsprotokolle sowie die Lernzielzusammenfassungen am Ende jeder Kurseinheit sehr hilfreich bei der Vorbereitung!

Bitte denkt daran nach Euren Prüfungen ein Gedächtnisprotokoll anzufertigen!

Prüfungsprotokoll
Theoretische Informatik A & B (1653/1654)

Prüfer: Prof. Dr. Weihrauch
Beisitzer: Prof. Dr. Heinemann
Datum: 16.09.2004 / 11.00
Dauer: 30min

- ***Wann heißt eine Zahlenfunktion berechenbar ?***
  Wenn es eine Registermaschine gibt die diese berechnet. Weitere Definitionen wurden nicht nachgefragt, was mich überraschte (obwohl ich „gewappnet“ gewesen wäre). Evtl. ist es nicht schlecht von sich aus auch die anderen beiden zu nennen (my-rekursiv und while-berechenbar).

- ***Was ist mit verallgemeinerten Registermaschinen ?***
  War etwas erschrocken, da gleich die zweite Frage „außerhalb“ von den Prüfungsprotokollen lag. Hab dann einfach gesagt, dass wenn die alle vorkommenden Tests und Funktionen dieser verallg. Registermaschine berechenbar sind, dass dann die „Gesamtfunktion“ auch berechenbar ist. Hier muss man ein wenig aufpassen, da im Script die verallg. Registermaschinen erst mal so definiert sind dass die Funktionen/Tests nicht berechenbar sein müssen.

- ***Wie sieht die Programmiersprache phi aus ?***
  Zuerst phi: N → P^(1) aufgeschrieben. Danach die 3 Funktionen aus denen phi sich zusammensetzt genannt und ganz grob beschrieben: also i-tes Bandprogramm, i-te Bandmaschine, i-te Zahlenfunktion.
  Dabei haben wir uns auch ein wenig unterhalten dass man hier sehr viel mit Verschlüsselungen arbeitet, denn „Band-„ hat ja zunächst mal nichts mit den einstellig-rekursiven Zahlenfunktionen zu tun. Herr Weihrauch wollte aber keine Definition von der Standardnummerierung ny.

- ***Welche Anforderungen stellt man vernünftige Programmiersprachen ?***
  Einfach die beiden Forderungen (U) und (S) informell von mir gegeben. Hab dann auch gesagt dass man diese beiden auch präzise mathematisch beschreiben kann mit Hilfe vom utm- und smd-Theorem. „Prompt“ kam die Frage:

- ***Definieren Sie mir doch bitte das utm- und das smn-Theorem !***
  Diese beiden habe ich „im Schlaf“ aufgeschrieben, da ich sie mir unendlich oft zur Übung aufgeschrieben habe. Dann noch ein paar Erläuterungen dazu gegeben. Schaden können solche Kommentare nie; Herr Weihrauch sagt dann schon von sich aus „das genügt mir schon“ !

- ***Es gibt da noch einen Satz zur effektiven Komposition. Kennen Sie den ?***
  Dieser fiel mir nach gut 10 sek ein….

- ***Wie lautet der Äquivalenzsatz von Rogers ?***
  Satz verbal aufgesagt. Wollte gerade ansetzen und erklären was es heißt „phi und psi sind äquivalent“ (mit dem Begriff Reduzierbarkeit), da sagte er wieder „das reicht schon“.

- ***Wann heißt eine Menge A Teilmenge N^(1) rekursiv ?***
  Habe die mit der charakteristischen Funktion genannt.

- ***Es gibt da noch eine mit dieser Umkehrung ?!***
  War überrascht, dass das Umkehrung gebrauchte, denn so wurde es natürlich einfach; Also heißt A rekursiv gdw. es eine total berechenbare Funktion f e R^(1) mit A=f^(-1)(0).

- ***Wann heißt eine Menge A Teilmenge N^(1) rekursiv-aufzählbar ?***
  Habe die mit den Definitionsbereichen von den berechenbaren Funktionen genannt.

- ***Nun kommt eine Frage die keinen Einfluss auf die Notenfindung hat. Können Sie den Unvollständigkeitssatz von Gödel, und sagen um was da geht ?***
  Es ist beruhigend , dass man vorher gesagt bekommt dies zählt quasi nicht zur Prüfung. Konnte dennoch die beiden Mengen U_x und V_x angeben (Projektionssatz) und ein wenig erläutern. Er hat dann noch von sich aus ein paar Sachen gesagt, die ich natürlich nicht verstanden hab *g* .
  Nach diesen Ausführungen meinte er dann zu mir „Sie haben diesen Satz so halbwegs verstanden“.

- ***Berechenbarkeit auf anderen Mengen: warum kann man die Menge der reellen Zahlen nicht so behandeln wie z.B. die Menge der rationalen Zahlen ?***
  R ist nicht mehr abzählbar.

- ***Was kann man da machen ?***
  Man verschlüsselt die rellen Zahlen durch unendliche Folgen von Symbolen über einem Alphabet (Darstellung). Habe dann „Sigma hoch w“ gesagt; natürlich heißt es „Sigma hoch Omega“.

- ***Welchen Typ Maschine verwendet man hier ?***
  Die Typ-2 Maschine.

- ***Warum rechnet die Typ-2 Maschine bei der Dezimaldarstellung falsch ?***
  Habe erklärt dass die Maschine falsch rechnet, wenn z.B. bei 3*3,3334 sie nur 2 Dezimalstellen nach dem Komma liest und sie somit 9,99 ausgibt zu einem Zeitpunkt t. Irgendwie hat dann Herr. Weihrauch herausgefunden dass ich die Typ-2 Maschine falsch verstanden habe. Ich ging davon aus dass diese Maschine bei jedem Lesen von den Eingabebändern gleich aufs Ausgabeband schreiben würde. Das tut sie nicht ! Sie liest endlich viele Zeichen von den Eingaben und erzeugt dann eine Ausgabe. Herr Weihrauch erkannte mein Missverständnis, indem er sich in mich hineinversetzte ! Er sagte „Jetzt verstehe ich warum **Sie** das nicht verstehen !“. Hat aber der Note nicht geschadet, weil im Endeffekt auch meine Schlussfolgerung dazu führt dass man mit der Dezimaldarstellung nicht auf der Typ-2 Maschine rechnen kann.

- ***Gibt es denn eine andere Darstellung der reellen Zahlen, um diesem Mangel zu beheben ?***
  Ja, die Cauchy-Darstellung. Diese erklärt über die unendlichen Folgen der ny-rat-Notation und dass diese sehr schnell gegen x konvergieren.

- ***Gut. Dann noch kurz etwas zur Komplexitätstheorie. Kennen Sie NP-vollständige Probleme ?***
  Habe 2-D-Domino und 3SAT genannt. Herr Weihrauch meinte dann, dass die Informatiker ein Patent darauf haben und der private Sender 3SAT sich diesen Begriff einfach „geschnappt“ hätten !

- ***Wie ist denn NP-vollständig definiert ?***
  Siehe Script.

- ***Und NP-hart ?***
  Siehe Script.

- ***Was bedeutet L<=pol M ?***
  Dass für alle x gilt : x e L ←→ f(x) e M , für ein f e FP

Fazit

Prof. Weihrauch ist wirklich ein absolut netter und fairer Prüfer !! Er leitet jede Frage mit ein paar Sätzen ein, sodass man sehr schnell weis, wo man sich im Stoff befindet. Er schafft eine sehr entspannte Atmosphäre und gibt immer ein positives Feedback , wenn man die Frage zu seiner Zufriedenheit erklärt hat.
Ich bin mir ziemlich sicher, dass er spätestens nach 5min weis, ob man sich mit dem Stoff intensiv auseinander gesetzt hat. Auch würde ich jedem raten bei Definitionen diese nicht nur aufzusagen, sondern etwas „schmückendes“ Beiwerk von sich aus zu sagen. Das spart Zeit. Prof. Weihrauch sagt dann schon von sich aus, dass das nun genüge.
Da bei mir auch 2-3 Fragen drankamen die nicht in den vorherigen Protokollen waren, ist anzuraten sich nicht ausschließlich auf die Protokolle zu verlassen. Deshalb gibt’s m.E. nach nur eins: LERNEN ! Man muss nicht bis ins letzte Detail alles verstanden haben, aber die Schlüsselelemente im Script sollten sitzen.
Wer also etwas getan hat, der muss sich bei dieser Prüfung wirklich keine Sorgen machen !! Auch die Benotung ist sehr wohlwollend. Ich hatte wirklich nicht mit einer 1,0 gerechnet wegen ein paar Fehlern.
Unvorbereitet in diese Prüfung zu gehen und zu hoffen dass Protokollfragen drankommen halte ich für falsch.

Ich hoffe ihr konntet ein wenig aus diesem Protokoll für Euch herausziehen und es ist für uns alle hilfreich wenn auch ihr diese Protokoll-Tradition fortführen werdet !

Alle die diese Prüfung noch vor sich haben viel, viel Erfolg ☺

Prüfer: Prof. Weihrauch
Beisitzer: Dr. Osterloh
Datum: 29.07.2004
Dauer: ca. 25-30 Minuten

Die Prüfung begann, wie eigentlich bei Prof. Weihrauch schon gewohnt mit der üblichen Einführung, was die theoretische Informatik so an Stoff bietet. Die erste Frage dann auch "wie immer":

(1) Welche Definitionen von berechenbaren Zahlenfunktionen kennen sie?
— Registermaschine, $\mu-$rekursive Fkt. und WHILE-Programme

(2) Was genau sind denn die $\mu-$rekursiven Funktionen?
— Menge Gr und Operatoren $Sub, Prk, \tilde{\mu}(h)$
Nach kurzer Überlegenszeit, was denn nun gefragt werden könne, kam:

(3) Es gibt ja berechenbare Funktionen. Kennen Sie auch eine nicht berechenbare Funktion?
— $g(i) = \begin{cases} \varphi_i(i) & falls\ \varphi_i\ existiert \\ 0 & sonst \end{cases}$ inkl. Beweis (er hatte anscheinend eine etwas andere Funktion im Sinn...)

(4) Dann nennen Sie doch mal die zwei wesentlichen Merkmalen einer Programmiersprache.
— (U) und (S) Eigenschaft von Anfang KE 5 zitiert.

(5) Was ist denn dann die Nummerierung φ?
— Erläutert anhand von $\varphi_i = \xi \circ \nu_M \circ \nu_P(i)$ inkl. i-tes Bandprogramm, i-te Bandmaschine, und $\xi(M) = \iota f_M \iota^{-1}$.

(6) Was ist denn die (U)-Eigenschaft von φ? Das utm-Theorem?
— Erläuterung, genaue Wiedergabe der berechenbaren Funktion $u_\varphi(i, x)$. Das smn-Theorem hat er danach nicht mehr fragen wollen.

(7) Besondere Eigenschaft von φ? Diese Programmiersprache ist ja eigentlich eher etwas willkürlich... Er wollte auf den Äquivalenzsatz von Rogers hinaus.
— Äquivalenzsatz erläutert, vorher kurz gesagt, dass φ bis auf Äquivalenz eindeutig bestimmt ist.

(8) Dann gibts da noch so einen ganz wichtigen Satz bzgl. der Programme.
— Satz von Rice erläutert, kurz gehangen, ob nun nicht rekursiv oder nicht r.a. aber ging gut.

(9) "Sternchenfrage": Anwendungen in der Logik inkl. Gödelschem Unvollständigkeitssatz etc. (hatte übrigens keinerlei Einfluss auf die Note)
— Hab ein bißchen was zu Beweissystemen und den daraus ableitbaren Sätzen erzählt, kam aber bei dem Nachweis, wie genau die Menge U_x jetzt nochmal aussieht durcheinander. Er hat das dann aufgeklärt.

(10) Was ist eine rekursive Menge?
— Zwei Definitionen mit charakteristischer Funktion und $f^{-1}(\{0\})$ hat ihm gereicht.

(11) Was ist eine rekursiv-aufzählbare Menge?
— Definitionsbereiche von partiell-rekursiven Funktionen. Eine rekursive Menge ist immer rekursiv-aufzählbar. Und entweder die Menge A ist leer oder aber sie ist Bild einer total-rekursiven Funktion.

(12) Warum ist die leere Menge rekursiv-aufzählbar?
— Hat etwas gedauert, musste länger überlegen und dusseliges Zeugs erzählen, aber letztlich liegts daran, dass es eine berechenbare Funktion gibt mit $Def(g) = \emptyset$.

(13) Kennen Sie eine Menge, die rekursiv-aufzählbar aber nicht rekursiv ist?
— K_φ nennen und kurz erklären, was es enthält.

Nun Teil B: Zuerst wurde von Prof. Weihrauch wieder so einiges erzählt, worum es da so ging. Dann die Fragen.

(14) Wie erkennt eine Kontrollturingmaschine, dass y ein Beweis für x ist? Was ist die Sprache der KTM?
— $L_M := \left\{x \in \Sigma^* \;\middle|\; \exists y \in \{0,1\}^*.\tau_M(x,y) = 1\right\}$

(15) Sie kennen doch bestimmt das Problem RUCKSACK. Was ist das?
— Es gibt eine Menge von Gegenständen mit den Gewichten $a_1, \ldots, a_n$ und eine Rucksack, der ein Gewicht N tragen kann. Dieser soll so gefüllt werden, dass gilt $\sum_{i \in M} a_i = N$. M ist dabei die Menge der ausgewählten Gegenstände.

(16) Kann man das RUCKSACK-Problem auch mit einer deterministischen Turingmaschine lösen?
— Irgendwie fand ich, es war ne Fangfrage. Habe ziemlich lange für die richtige Antwort gebraucht. Es geht, denn es gilt ja die Abschätzung $P \subseteq NP \subseteq PSPACE \subseteq \bigcup_{c \in \mathbb{N}} ZEIT(2^{n^c})$. Also benötigt die deterministische Turingmaschine nur ziemlich viel ZEIT dafür...

(17) Wie ist die Regelmenge bei rechtslinearen Grammatiken?
— $R \subseteq \Pi^1 \times \Sigma^* \cup \Sigma^* \cdot \Pi^1$

(18) Und bei Normalformgrammatiken?
— $R \subseteq \Pi^1 \times \Sigma^1 \cdot \Pi^1 \cup \{\varepsilon\}$

(19) Wie ist das bei kontextfreien Grammatiken mit der Regelmenge?
— $R \subseteq \Pi^1 \times (\Sigma \cup \Pi)^*$

(20) Zum Abschluss: wie lassen sich die regulären Ausdrücke aus den endlichen Automaten ermitteln?
— Rekursionsformel für L_{ij}^k

Mein Fazit:
Also, Prof. Weihrauch ist irgendwie echt der beste Prof, den ich bisher getroffen habe (auch wenns noch nicht so viele waren). Zuerst dachte ich immer, alle anderen hätten in ihren Prüfungsprotokollen stark übertrieben, aber er ist wirklich so nett. Ich weiß nicht, wie er es macht, aber er schafft es, einem die Angst vor der Prüfung zu nehmen. Viel Erfolg euch anderen bei den Prüfungen.... kann Prof. Weihrauch nur empfehlen!!

Prüfungsprotokoll Diplomvorprüfung Theoretische Informatik Kurse 01653/01654

Prüfer: Prof. Weihrauch
Februar 2004
Note 1,7

- Wann ist eine Funktion f berechenbar? --- Wenn es eine Registermaschine gibt, die f berechnet.
- Definition rekursive Menge
- Gibt es auch nicht berechenbare Funktionen? - Ja
- Kennen Sie eine? Hier habe ich erst gegrübelt, beim Hinweis auf Diagonalbeweis konnte ich dann aber eine konstruieren (f (i) = div falls phi (i) existiert, sonst 0) und gleich das Verfahren beim Diagonalbeweis erläutert
- Warum ist K phi rekursiv aufzählbar?
- Zusammenhang zwischen rekursiv und rekursiv aufzählbar
- "Es gibt einen Satz A Teilmenge von P^1 ... nicht rekursiv" Hier war der Satz von Rice gemeint, den habe ich dann erläutert
- Definition rekursiv aufzählbar
- Ist die leere Menge rekursiv aufzählbar? - Ja
- Warum? Können Sie das an der Definition von "rekursiv aufzählbar" festmachen? - Hier weiss ich bis heute nicht genau, wie das gemeint war, wir haben zumindest über Maschinen ohne Eingabe gesprochen, die in Endlos-Schleifen laufen

- Was bedeutet FZEIT(n) ?
- Def. tm schlange (n), also maximale Rechenzeit von Worten der Länge n erklärt
- Satz von Savitch erläutert
- Wie ist der Zusammenhang zwischen Rechenzeit von nichtdeterministischen Funktionen und deterministischen? ==> exponentiell, wußte ich aber nicht
- Wie kommt man auf die Zustandsmenge der Potenzautomaten? -- Hier habe ich das Verfahren erklärt
- Welche Zustände von Potenzautomaten sind Endzustände? - Diejenigen , die Endzustände vom Ursprungsautomaten enthalten.

Die Prüfung verlief in ruhiger und freundlicher Atmosphäre. Bei einigen Fragen wußte ich allerdings nicht gleich immer, wie sie gemeint waren. Prof. Weihrauch hat dann aber nochmals erklärt und nachgefragt, wenn meine Antworten nicht ausreichend waren.
Insgesamt kann ich Professor Weihrauch als Prüfer empfehlen.

Viel Erfolg bei Eurer Prüfung!

Vordiplomsprüfung Theoretische Informatik A und B
Gedächtnisprotokoll

1)
Prüfer: Prof. Dr. Weihrauch
Beisitzer: Prof. Dr. Heinemann
Zeit: 13.10.2003, 09:30 Uhr
Dauer: ca. 30 Minuten
Note: nicht bestanden

2)
Prüfer: Prof. Dr. Weihrauch
Beisitzer: Dr. Daren Kunkle
Zeit: 05.01.2004, 10:00 Uhr
Dauer: ca. 30 Minuten
Note: 1,3

Wie ihr seht musste ich zweimal ran. Daher das (für mich) Wichtigste zuerst:
Es genügt nicht, den Stoff nur weitgehend verstanden zu haben. Die geforderten Sätze und Definitionen muss man auch selbständig (möglichst exakt) formulieren können.
Ich war vor meiner ersten Prüfung so glücklich darüber, dass ich den Inhalt weitgehend verstanden habe, dass ich darauf vertraute während der Prüfung verbal alles schon richtig herauszubekommen. Ein Irrtum.
Ein Tipp: Wenn ihr, wie ich, keine Möglichkeit habt, euch mit Kommilitonen oder einen Dozenten über den Stoff zu unterhalten, legt euch ein Diktiergerät zu. Fragt euch selbst ab und hört euch eure Lösungen mal an. Bei mir hat es geholfen.
Bezüglich der Beurteilung der Prüfung und des Prüfers kann ich mich meinen Vorgängern nur anschließen. Absolut empfehlenswert (das Vergeigen der ersten Prüfung geht voll auf meine Kappe).
Auch wenn die Fragen alle schon mal da waren, zur Vervollständigung:

Erste Prüfung:

1. Was sind die berechenbaren Zahlenfunktionen
2. Was sind die berechenbaren Wortfunktionen (Band <-> Touring)
3. Was versteht man unter ‚nennt man berechenbar' und ‚ist berechenbar'
4. Multiplikation berechenbar ?
5. Churchsche-These
6. Zusammenhang von berechenbarer Wort- und Zahlenfunktion
7. Standardnummerierung phi: phi : N -> ?
8. utm-Theorem
9. Definition von rekursive Menge (charakt. Funktion – für jede Menge gibt es diese Funktion ?)
10. Definition von r.a. Menge
11. Nennen sie eine r.a. Menge: Beweisen Sie, dass sie r.a. aber nicht rekursiv ist.
12. Gibt es ‚Nicht r.a. Mengen' ? Nennen sie eine (mit Definition).
13. Berechenbarkeit anderer Mengen als Zahlen – ny sei eine Nummerierung; g eine Funktion. Was bedeutet ny-berechenbar ?
14. Definieren Sie das Pumping-Lemma für kontextfreie Sprachen.
15. Nennen sie eine kontextfreie Sprache und deren Regel.
16. Nennen sie eine Sprache, die nicht kontextfrei ist.
17. Kontrolltouringmaschine: L_M = { ...

Zweite Prüfung:

1. Wann ist eine Zahlenfunktion berechenbar
2. Wann ist eine Wortfunktionen berechenbar
3. Mathematische Definition einer Registermaschine
4. Zusammenhang von berechenbarer Wort- und Zahlenfunktion
5. Eine nicht berechenbare Funktion nennen und begründen
6. Standardnummerierung phi.
7. Vorraussetzungen für eine gute Programmiersprache
8. smn-Theorem alleine auch schon ausreichend (effekt. Komposition)
9. Wann ist eine Menge rekursiv
10. Definition von r.a. Menge
11. Zusammenhang rek. und r.a. Menge
12. Nennen sie eine r.a. aber nicht rekursive Menge.
13. Gibt es ‚Nicht r.a. Mengen' ? Nennen sie eine (mit Definition).
14. Berechenbarkeit anderer Mengen als Zahlen – Wann ist eine Menge ny-berechenbar ?
15. Wie ist die Nummerierung definiert ?
16. Definieren Sie das Pumping-Lemma für kontextfreie Sprachen.
17. Regel für eine rechtslineare Sprache
18. a^n b^m - Begründung
19. ww^r
20. a^n b^n c^n
21. Was bedeutet linkslineare Sprache und wie unterscheidet sie sich zur rechtslinearen
22. Wieso kein Unterschied bez. Sprache
23. Zusammenhang Band – Zeit mit Begründung
24. GAP (Nlogspace – was ist überhaupt GAP)
25. Nlogspace-vollständig (Bedeutung)
26. nlogspace-hart (Bedeutung)

Vordiplomprüfung Theoretische Informatik A und B
Gedächtnisprotokoll

Prüfer: Prof. Dr. Weihrauch
Beisitzer: Dr. Vasco Bratka
Zeit: 21.11.2003, 12:00 Uhr
Dauer: ca. 30 min
Note: 1,0
Inhalte: 01653 (Version WS 2002/2003), 01654 (Version SS 2002)

TEIL A

- **Wann heißt eine Zahlenfunktion berechenbar?**
- **Churchsche These erklären**
- **Wann heißt eine Wortfunktion berechenbar?**
- **Warum brauchen wir Bandmaschinen?**
- **Wir hatten berechenbare Zahlen- und berechenbare Wortfunktionen. Gibt es da einen Zusammenhang?**
- **Was ist φ?**
- **Utm und smn-Theorem aufschreiben und erklären**
- **Da gibt es noch ein Satz mit smn und effektiven Komposition. Wie lautet der ?**
- **Wann ist eine Menge rekursiv, wann r.a ?**
- **Welche Menge ist r.a aber nicht rekursiv ?**
- **Beweisen dass $K\varphi$ nicht rekursiv ist (genau erklären)**
- **Welche Rolle spielen r.a. Mengen in der Logik?**
- **Wird am Computer mit den reelen Zahlen gerechnet (Nein, weil die Menge R nicht abzählbar ist)**
- Darstellung der realen Zahlen, Cauchy Darstellung,Typ-2-Maschine erklären
- Warum kann man die dezimal Darstellung nicht nehmen?
- Können sie beweisen dass 3*n nicht berechnbar (das konnte ich nicht)
- Wie lautet der fundamentaler Satz der Algebra?

TEIL B

- **Was sind reguläre Sprachen?**
- **Wann wird ein Wort von einem nichtdeterminierendem Automat erkannt?**
- **Welche NP-Vollst. Probleme kennen Sie?**
- **Was ist GAP ?**
- **Zu welcher Komplexitätsklasse gehört GAP? ->NLOGSPACE**
- **Was bedeutet das? Es gibt eine KTM, die das GAP-Problem mit O(log) Zeitbedarf löst.**
- **Wie lautet der Satz von Savitch ?**

Der Prof. Weihrauch ist sehr netter und ruhiger Professor, der als Prüfer auf jedem Fall zu empfehlen ist.
Bei einigen Fragen (wie zB beim $K\varphi$ Beweis) wollte er genaue Erklärung haben, um zu sehen ob man es auch verstanden hat.
Bei der Vorbereitung habe ich alle Prüfungsprotokolle durchgearbeitet und die Fragen die bei mir kamen, waren fast alle schon mal drann.

Viel Glück bei euerer Prüfung!

Vordiplomprüfung Theoretische Informatik A und B

Gedächtnisprotokoll

Prüfer: Prof. Dr. Weihrauch
Beisitzer: Priv.-Doz. Dr. Heinemann
Zeit: 28.02.2003, 10:00 Uhr
Dauer: ca. 25 min
Note: 1,0
Inhalte: 01653 (Version WS 2001/2002), 01654 (Version SS 2002)

Theoretische Informatik - da hatten wir zwei Kurse ...
Prof. Weihrauch leitete die Prüfung ein, indem er in wenigen Sätzen die Inhalte der beiden Kurse zusammenfasste.

Beginnen wir mit Teil A: Wann heißt eine Zahlenfunktion berechenbar?
Eine Zahlenfunktion heißt berechenbar, genau dann
- wenn es eine Registermaschine gibt, die diese Funktion berechnet,
- wenn die Funktion WHILE-berechenbar ist,
- wenn die Funktion μ–rekursiv ist.

Wann heißt denn eine Funktion μ-rekursiv?
Eine Funktion heißt μ–rekursiv, genau dann wenn sie sich in endlich vielen Schritten aus der Menge Gr durch wiederholtes Anwenden der Operatoren Prk, μ und Sub erzeugen lässt. Prk, μ und Sub sind die Operatoren der primitiven Rekursion, der μ–Rekursion und der Substitution. Das Aufschreiben der Menge Gr war nicht erforderlich.

Was bedeutet es, wenn wir sagen: eine Funktion "heißt" berechenbar?
Mit "... heißt berechenbar ..." meint man, dass die Funktion gemäß unserer mathematischen Definition (Registermaschine) berechenbar ist. Man kann sich leicht klar machen, dass man zu jeder Registermaschine ein ASSEMBLER- oder PASCAL-Programm angeben kann, was die Berechnung der Maschine auf einem realen Rechner nachvollzieht. Insofern kann man sagen, dass jede Register-berechenbare Funktion auch maschinell berechenbar ist. Es stellt sich die Frage, ob es maschinell berechenbare Funktionen gibt, die nicht Register-berechenbar sind. Bis heute ist es nicht gelungen, eine maschinell berechenbare Funktion anzugeben, die nicht Register-berechenbar ist. Nach der *Churchschen These* sind die Register-berechenbaren Funktionen *genau* die maschinell berechenbaren Zahlenfunktionen.

Welche Maschinenmodelle haben wir im Zusammenhang mit Wortfunktionen kennen gelernt?
Turingmaschinen und Bandmaschinen.

Wann heißt eine Wortfunktion berechenbar?
Eine Wortfunktion heißt berechenbar, genau dann wenn es eine Turingmaschine gibt, die diese Funktion berechnet.

Wir hatten also berechenbare Zahlen- und berechenbare Wortfunktionen. Gibt es da einen Zusammenhang?
Der Zusammenhang wird hergestellt durch bijektive Standardnumerierungen $\nu : \mathbb{N} \to \Sigma^*$. Mit einer Standardnumerierung ν kann man Wörter durch Zahlen und umgekehrt Zahlen durch Wörter verschlüsseln. Hinsichtlich der Berechenbarkeit gilt Folgendes (ich brauchte es nur für einstellige Funktionen zu zeigen): Eine Wortfunktion $g :\subseteq \Sigma^* \to \Sigma^*$ ist berechenbar, genau dann wenn ihre Verschlüsselung $\nu^{-1} g \nu :\subseteq \mathbb{N} \to \mathbb{N}$ eine berechenbare Zahlenfunktion ist.

Wann heißt eine Menge A Í IN rekursiv?
Ich habe die vier im Kurs genannten Charakterisierungen aufgezählt:
- Eine Menge heißt rekursiv, gdw. ihre charakteristische Funktion berechenbar ist.
- Eine Menge $A \subseteq \mathbb{N}$ ist rekursiv, gdw. es eine totale berechenbare Funktion $f \in R^{(1)}$ gibt mit $A = f^{-1}\{0\}$.
- Es sei $A \subseteq \mathbb{N}$ *unendlich*. Dann gilt: A ist rekursiv, gdw. es eine *wachsende* Funktion $f \in R^{(1)}$ gibt mit $A = \mathrm{Bild}(f)$.
- Eine Menge $A \subseteq \mathbb{N}$ ist rekursiv, gdw. A und das Komplement $\mathbb{N} \setminus A$ rekursiv-aufzählbar sind.

Gut, dann will ich nach der Definition von rekursiv-aufzählbar nicht mehr fragen.
Wie zeigt man denn, dass eine Menge, die rekursiv ist, auch rekursiv-aufzählbar ist?
Den Beweis sollte ich auf Maschinenebene führen. Prof. Weihrauch gab den Tipp, erst einmal zu erklären, was rekursiv bzw. rekursiv-aufzählbar auf der Ebene der Maschinen bedeutet:

- Wenn eine Menge $A \subseteq IN$ rekursiv ist, gibt es eine Registermaschine, die für jedes $x \in IN$ die Antwort "ja" liefert, falls $x \in A$ gilt, und die Antwort "nein" sonst. Die Maschine hält immer. Sie berechnet die charakteristische Funktion.
- Wenn eine Menge $A \subseteq IN$ rekursiv-aufzählbar ist, gibt es lediglich eine Registermaschine, die für $x \in IN$ die Antwort "ja" liefert, wenn $x \in A$ gilt, und im Fall $x \notin A$ *keine* Antwort liefert. Hier spiegelt sich wieder, dass die rekursiv-aufzählbaren Mengen genau die Definitionsbereiche der partiell berechenbaren Funktionen sind.

Beweis: Sei A rekursiv. Dann gibt es eine Maschine, die die charakteristische Funktion berechnet (s. o.). Diese Maschine muss nun so abgewandelt werden, dass sie bei dem Ergebnis "nein" ($x \notin A$) nicht anhält, sondern in einer Endlosschleife verweilt. Die Maschine berechnet dann eine partiell berechenbare Funktion f mit A = Def(f). Somit ist A auch rekursiv-aufzählbar.

Wenn eine Menge rekursiv-aufzählbar ist und ihr Komplement ist es auch:
Wie zeigt man, dass diese Menge auch rekursiv ist?
Hier konnte ich mir aussuchen, ob ich es auf der Ebene der Maschinen oder auf der Ebene der Funktionen erkläre.
Ich habe mich für die Funktionsebene entschieden: Der Beweis beruht auf dem Satz, dass eine Menge $A \subseteq IN$ rekursiv-aufzählbar ist, gdw. $A = \emptyset$ oder A = Bild(g) für ein $g \in R^{(1)}$. Da also die Mengen A und $IN \setminus A$ rekursiv-aufzählbar sind, gibt es Funktionen $f, g \in R^{(1)}$ mit A = Bild(f) und $IN \setminus A$ = Bild(g). Somit muss eine Zahl $y \in IN$ in *einer der beiden Listen* (f(0), f(1), ...) und (g(0), g(1), ...) vorkommen. Man konstruiert nun eine Funktion, die die beiden Listen parallel nach dieser Zahl durchsucht. Wird die Zahl in der Liste Bild(f) = A gefunden, gibt die Funktion eine 1 aus, ansonsten (wenn also die Zahl in Liste Bild(g)) eine 0. Die so konstruierte Funktion ist berechenbar. Offenbar handelt es sich um die charakteristische Funktion der Menge A. Daraus folgt: A ist rekursiv.
Prof. Weihrauch: "Sie hätten auch zwei Maschinen parallel laufen lassen können." (Hm, diese Beweisvariante wäre vielleicht einfacher/kürzer gewesen. Man lässt die Maschinen laufen, solange bis eine der beiden hält. Das entspricht dem parallelen Durchsuchen der Listen. Im Kurstext wird der Beweis auf Funktionsebene geführt. Deshalb war mir das zunächst vertrauter. Wenn ich es noch mal zu beantworten hätte, würde ich es auf Maschinenebene erklären.)

Welche Rolle spielen die rekursiv-aufzählbaren Mengen in der Logik?
Die Mathematik ist axiomatisch aufgebaut. Jedes in der Mathematik gebräuchliche Beweissystem (Axiomensystem) ist eine rekursive Menge. Die Mengen, der daraus ableitbaren (beweisbaren) wahren Sätze sind rekursiv-aufzählbar. Aus dem Namen einer solchen Menge kann ein weiterer wahrer Ausdruck hergeleitet werden. Die so erzeugten Ausdrücke sind *nicht* aus den Axiomen herleitbar! Daraus ergibt sich der erste Gödelsche Unvollständigkeitssatz, dass jedes widerspruchsfreie rekursive Axiomensystem unvollständig ist.

Im Kurs wurde die Programmiersprache j eingeführt. Was ist denn Kj ?
Ich habe die Menge $K\varphi = \{i \in IN \mid \varphi_i(i) \text{ existiert}\}$ aufgeschrieben und gesagt, dass sie rekursiv-aufzählbar, aber nicht rekursiv ist.

Der Beweis, dass Kj rekursiv-aufzählbar ist, ist ja einfach: Man macht es mit Hilfe der universellen Funktion. Aber wie zeigt man, dass Kj nicht rekursiv ist?
Es ist zu zeigen, dass $IN \setminus K\varphi$ nicht rekursiv-aufzählbar ist. Man zeigt es durch Diagonalisierung über die Folge *aller* rekursiv-aufzählbaren Teilmengen von IN - also über ($Def(\varphi_0)$, $Def(\varphi_1)$, ...).
Hierzu habe ich die folgende Äquivalenzbeziehung aufgeschrieben: $i \in IN \setminus K\varphi \Leftrightarrow i \notin K\varphi \Leftrightarrow i \notin Def(\varphi_i)$
Daraus folgt, dass die Mengen $IN \setminus K\varphi$ und $Def(\varphi_i)$ ungleich sind für alle $i \in IN$. Die Menge $IN \setminus K\varphi$? stimmt also mit *keiner* rekursiv-aufzählbaren Menge überein und ist daher nicht rekursiv-aufzählbar. Weil das Komplement von $K\varphi$ nicht rekursiv-aufzählbar ist, ist $K\varphi$ nicht rekursiv.

Gut, dann kommen wir mal zur Komplexitätstheorie ...
Wir hatten den Zeitbedarf $t_M(x)$ und den Bandbedarf $s_M(x)$. Hängen die irgendwie zusammen?
Ich habe erklärt, was man unter Time-Space-Tradeoff versteht, und die Abschätzungsformeln aufgeschrieben. Außerdem sollte ich begründen, wie es zu den Abschätzungen kommt:

- Für alle $x \in Def(f_M)$ ist $s_M(x) \leq t_M(x) + k$.

k ist der Bandbedarf der Anfangskonfiguration (zu Beginn ist auf jedem der k Arbeitsbänder genau ein Feld belegt - nämlich das Blank unter dem Kopf). $t_M(x)$ ergibt sich daraus, dass man pro Rechenschritt (durch Anfügen eines Blanks) *höchstens* ein neues Feld hinzugewinnen kann.

- Es gibt ein $c \in \mathbb{N}$, so dass $t_M(x) \leq (\lg(x) + 1) \cdot c^{sM(x)}$ für alle $x \in \mathrm{Def}(f_M)$.

Die zweite Abschätzung hat etwas mit der Anzahl der möglichen Konfigurationen zu tun: Die Berechnung liefert bei Eingabe von x eine Folge von Konfigurationen. Bei einer haltenden Berechnung können zwei verschiedene Konfigurationen *nicht äquivalent* sein, sonst käme man nie aus der Berechnung heraus und hätte eine Endlosschleife! Die zweite Formel ergibt sich, wenn man die Anzahl der Äquivalenzklassen von Konfigurationen nach oben abschätzt.

Noch zur zweiten Formel: Warum ist die Abschätzung exponentiell?
Bei einem Band der Länge s und einem Alphabet mit g Zeichen (also $g := \mathrm{card}(\Gamma)$) gibt es g^s Möglichkeiten, dieses Band zu beschreiben. Mehr wollte Prof. Weihrauch nicht hören und meinte: "Das reicht schon. Das ist die Stelle im Beweis, an der das Exponentielle reinkommt".

Was sind P und NP?
P ist die Menge aller in polynomieller Zeitkomplexität *deterministisch* entscheidbaren Sprachen.
NP ist die Menge aller in polynomieller Zeitkomplexität *nichtdeterministisch* beweisbaren Sprachen.
Weil jeder deterministische Algorithmus ein Sonderfall eines nichtdeterministischen (mit nur einer Möglichkeit) ist, gilt: $P \subseteq NP$. Aber ob P = NP gilt, ist bisher noch nicht geklärt und ist eines der zentralen Probleme der theoretischen Informatik.

Nennen Sie mir ein Maschinenmodell, was wir im Zusammenhang mit den nichtdeterministischen Komplexitätsklassen kennen gelernt haben!
Kontroll-Turingmaschine (KTM).

Wann wird denn ein Wort von einer Kontroll-Turingmaschine erkannt?
Ein Wort x wird von der KTM erkannt, genau dann wenn es mindestens ein Hilfseingabewort $y \in \{0, 1\}^*$ gibt, so dass die Maschine bei Eingabe von (x, y) zu einer positiven Entscheidung gelangt (wenn also y ein Beweis für die Eigenschaft "$x \in L$" ist). Das Hilfseingabewort steht auf Band 0.
Auch hier reichte die verbale Umschreibung. Die Menge $L_M = \{x \in \Sigma^* \mid \exists\, y \in \{0, 1\}^*.\tau_M(x, y) = 1\}$ musste ich nicht aufschreiben.

Nehmen wir mal 3SAT. Was steht hier bei der KTM auf Band 0?
Die Belegung (Werte) der Variablen. Die KTM prüft dann in polynomieller Zeit, ob die Formel bei dieser Belegung erfüllbar ist oder nicht.

Hier war die Prüfung zu Ende. Formale Sprachen wurden nicht geprüft.

Fazit: Ich kann mich den bisherigen Prüfungsprotokollen nur anschließen. Prof. Weihrauch ist als Prüfer uneingeschränkt zu empfehlen. Die Prüfungsatmosphäre war sehr freundlich. Es war eher ein Gespräch als ein Frage-Antwort-Spiel. Prof. Weihrauch gestaltet die Prüfung wie eine Kurszusammenfassung. Die Fragen ergeben sich immer aus dem Kontext. Wenn man die Zusammenhänge verstanden hat, kann eigentlich nichts schief gehen. Es ist auch nicht schlimm, wenn mal eine falsche Antwort herausrutscht (mir passiert bei 3SAT). Wenn man den Fehler noch selbst korrigieren kann, hat es keine negativen Konsequenzen.

Vorbereitung: Schon zur Vorbereitung auf die Klausur habe ich die Betreuungsangebote in den Studienzentren wahrgenommen. Wer irgendwie die Möglichkeit hat, an solchen Veranstaltungen teilzunehmen, sollte dies zu tun. Dabei lernt man vor allem, über den Stoff zu reden.
Viel gebracht haben mir auch die vom Lehrgebiet Theoretische Informatik I veranstalteten *Studientage*. Diese sind auch als Vorbereitung auf die mündliche Prüfung sehr gut geeignet. Dort wurde zu jedem Themenschwerpunkt ein kurzer Vortrag gehalten. Anschließend wurden in Kleingruppen die zugehörigen Aufgaben besprochen.
Die Schwerpunkte der mündlichen Prüfung kann man gut anhand der schon vorhandenen *Gedächtnisprotokolle* erkennen. Grundlage für meine Vorbereitung war zunächst das Lernskript von Thomas Schwarze (siehe „Rund ums Fernstudium“). Das habe ich dann anhand der jüngeren Protokolle noch etwas ergänzt. Anschließend habe ich den Kurstext mit Blick auf die am Ende einer jeden KE genannten Lernziele durchgearbeitet und mir das, was noch in meinen Aufzeichnungen fehlte, herausgeschrieben.

Wer die Klausur bestanden hat (und das haben ja alle, die zur mündlichen Prüfung fahren :-)), der muss sich hier wirklich keine Sorgen machen.

Viel Glück bei Euren Prüfungen !!!

Prüfungsprotokoll - Diplomvorprüfung Theoretische Informatik

Prüfer :	Prof.Dr.Weihrauch
Beisitzer :	Priv.Doz. Dr. Heinemann
Datum:	16.09.2002
Dauer:	ca.30min
Note :	1.3

- ⇨ Wann heißt eine Zahlenfunktion berechenbar?
- ⇨ Wir haben im Kurs noch 2 weitere, äquivalente Definitionen kennen gelernt – welche waren das?
- ⇨ Wann nennt man denn eine Zahlenfunktion μ - rekursiv ?
- ⇨ Ist eigentlich die Multiplikation berechenbar ?
- ⇨ Wir haben im weiteren auch berechenbare Wortfunktionen betrachtet –wann ist denn eine Wortfunktion berechenbar?
- ⇨ Welche Zusammenhang besteht denn da zu den b.b. Zahlenfunktionen?
- ⇨ Haben unsere Berechenbarkeitsbegriffe irgendetwas mit tatsächlichem Rechen zu tun (z.B. mit einem Digitalrechner)? → *Chursche These !*
- ⇨ Wir haben dann im weiteren die Turingmaschine vereinfacht – in welcher Form? → *Bandmaschine*
- ⇨ Auch bei der Bandmaschine haben wir eine Vereinfachung in Bezug auf das Bandalphabet vorgenommen – was wissen Sie darüber? → *Satz ‚Für jede BANDMASCHINE M gibt es eine BM M' mit …..'*
- ⇨ Mit diesen Vereinfachungen war es nun leichter, f (phi) zu definieren. Was genau ist denn f? → *Nummerierung aller bb Zahlenfunktionen / Modellprogrammiersprache*
- ⇨ Was bezeichnet dabei f(i) ? → *Die i.berechenbare Funktion*
- ⇨ Wann ist eine Menge rekursiv? → *charakteristische Funktion bb*
- ⇨ Wann ist eine Menge A rekursiv aufzählbar ? → *Def.Bereich einer bb Funktion*
- ⇨ Kennen Sie noch eine andere Charakterisierung ? → *Projektionssatz*
- ⇨ Welcher Zusammenhang besteht zwischen den beiden Mengen ? → *A rek. -> A r.a.*
- ⇨ Und die Umkehrung ? → *Gilt i.Allgm. nicht*
- ⇨ Da gibt es noch einen weiteren Zusammenhang! → *A. r.a. und N/A r.a -> A ist rek.*
- ⇨ Sind diese Mengen gleich ? *(Hier habe ich zunächst ziemlich Wirres Zeug geredet, von dem ich dachte es sei richtig – war's wohl nicht! Prof.Weihrauch wollte auf K(phi) raus. Die richtige Antwort wäre also→ NEIN gewesen.)*
- ⇨ Gibt es denn eine Menge, die r.a. aber nicht rekursiv ist? → *K(phi)*

- ⇨ Wie zeigt man, dass diese Menge r.a. und nicht rek. ist? → *Den Beweis habe ich nur noch zum Teil gewusst. Man muss zeigen, dass die Menge r.a. ist- das ist leicht und er wollte es deswegen nicht sehen. Dann muss man zeigen, dass das Komplement nicht r.a. ist ....*

- ⇨ Gibt es Mengen, die weder r.a. NOCH rek. sind ? *Noch etwas angeschlagen von der Frage vorher habe ich die Menge Äq (Äquivalenzproblem) genannt und musste sie definieren. Einfacher wäre N/K(phi) von oben gewesen!*

- ⇨ Kennen Sie den Satz von Rice?

- ⇨ Komplexitätstheorie……..welcher Zusammenhang besteht denn zwischen dem Platz- und Zeitbedarf einer Maschine
- ⇨ Wie erklärt sich das ? Insbesondere, warum lässt sich der Zeitbedarf durch den Platzbedarf abschätzen ? → *Nur begrenzte Anzahl von Konfigurationen auf einem bestimmten Platz möglich……*

- ⇨ Warum ist die Abschätzung exponentiell ? → *Wusste ich nur zum Teil – mit Hilfe von Prof. Weihrauch hab ich's dann hinbekommen.*

- ⇨ Wir haben im Kursteil B das Graphenproblem GAP kennengelernt…(kurz erklärt)…..zu welcher Problemklasse zählt es ?*(Habe zuerst NP geantwortet – was falsch war. Prof.Weihrauch hat nochmals nachgefragt, etwas präziser und dann ist es mir noch eingefallen)* → *GAP ist LOGSPACE vollständig*

- ⇨ Was heißt LOGSPACE vollständig ?

- ⇨ Wann ist ein Problem LOGSPACE-hart?

- ⇨ Wir haben endliche determinierte und nichtdeterminierte Automaten kennen gelernt. Ein Wort, welches determiniert…….kurze Erklärung….. Wie sieht das für ein Wort aus, welches NICHTdeterminiert erkannt wird ?

- ⇨ Man kann nun für jeden endlichen Automaten einen vollständig determinierten endliche Automaten konstruieren. Wie macht man das? → *Potenzautomat*

- ⇨ Wie sieht da die Zustandsmenge aus? → *2 hoch Q*

- ⇨ Für was steht dieser Ausdruck ? → *Er wollte Potenzmenge hören, was mir nicht mehr Einfallen wollte – hab es dann versucht zu Erklären, wie ich es von meinem Mentor erklärt bekommen hatte …. Was wohl etwas seltsam rüber gekommen sein muss ☺*

Insgesamt war die Prüfung in ruhiger Atmosphäre und man bekommt auch genug Zeit zum Nachdenken. Prof. Weihrauch gestallte die Prüfung wie eine Zusammenfassung des gesamten Kurses und erklärt auch zwischen den Fragen viel selbst – was mir persönlich viel Nervosität genommen hat. Seine Orientierung am Kurstext war für mich auch beruhigend, weil man so ungefähr weiß, an welcher ‚Stelle' der Prüfung man sich befindet. Wie ich ja beschrieben habe, darf man auch ruhig mal eine falsche Antwort geben – gut ist's halt wenn man sie noch korrigieren kann ☺. Viel Glück bei Euerer Prüfung!

Mündliche Diplomvorprüfung Theoretische Informatik A+B

Prüfer: **Prof. Weihrauch**
Termin: 14.03.2002
Note: 1,3

Teil A:
Wann ist eine Zahlenfunktion berechenbar?
Kennen Sie weitere Definitionen? -> WHILE Programme, ? - Rekursion
Wie ist denn die ? - Rekursion definiert? -> grobe Beschreibung reichte
Wieso sagt man heißt berechenbar? -> Churchsche These
Wann ist eine Wortfunktion berechenbar?
Gibt es einen Zusammenhang zwischen der Berechenbarkeit von Zahlen- und Wortfunktionen?

Überleitung von Prof. Weihrauch zu ? -> Definition
Was sind denn hierbei die Programme? -> die natürlichen Zahlen
Die 2 grundlegenden Eigenschaften an dieser Stelle? -> utm und smn
Was ist denn das utm-Theorem? -> Definition und Beschreibung

Wie sieht es denn mit der Berechenbarkeit auf anderen Mengen aus? -> abzählbare Mengen, die reellen Zahlen als Beispiel für eine nicht abzählbare Menge, Darstellung der reellen Zahlen, Problem der nicht berechenbaren Multiplikation bei der Dezimaldarstellung, Cauchy Darstellung
Können Sie beweisen, dass die Multiplikation nicht berechenbar ist?
Kennen Sie den letzten Satz aus der KE 7 (fundamentale Eigenschaft der Algebra) -> war irgendwas mit Stetigkeit, ich wusste aber überhaupt nicht was er wollte

Teil B:
Welche Regeln gelten denn bei den rechtslinearen Grammatiken?
Die Regeln in der Normalform?
Wie kann man denn eine rechtslineare Grammatik in ihre Normalform überführen?
PASCAL ist ja durch eine kontextfreie Grammatik beschreibar (Backus Naur Form). Warum ist es dennoch nicht kontextfrei? -> Wegen Variablendeklaration, Beweis durch PumpingLemma

Überleitung zur Komplexitätstheorie:
Nehmen wie mal die Bandkomplexität. Kennen Sie GAP? -> naja, so ungefähr
Was ist denn das besondere an GAP? -> das man so wenig Band braucht (wusste ich aber nicht)
Wie sind denn P und NP definiert?
Was ist denn das N-NP Problem?
Definition von NP-vollständig?
Was bedeutet A aus NZEIT(n^3) genau? -> alle möglichen Pfade müssen in Zeit n^3 durchlaufen sein (hier wusste ich überhaupt nicht was er hören wollte)

Mein Eindruck von der Prüfung:
Prof. Weihrauch fängt die Prüfung immer gleich an. Das ist schon mal sehr nett. Und dann läßt er sich für die nächste Frage sehr viel Zeit, man merkt wie er überlegt, was er denn mal als nächstes fragen könnte. Ich denke man kann den Prüfungsverlauf sehr gut steuern, da er keinen festen Plan hat.
Und die Benotung ist sehr, sehr wohlwollend. Ich war von meiner guten Note doch echt überrascht.

Prüfungsprotokoll zur Vordiplomprüfung **„Theoretische Informatik“**

Prüfer: Prof. Dr. Weihrauch
Beisitzer: Herr Schröder
Datum: März 2002
Dauer: ca. 25 Minuten

Zuerst fand eine sehr freundliche und sympathische Begrüßung durch den Professor statt, der damit auch gleich einen Eindruck der Prüfungsatmosphäre vermittelte, der über das gesamte Gespräch erhalten blieb. Danach erfolgte eine kurze Einstimmung in die Themengebiete der beiden zu prüfenden Kurse, die er ebenfalls selbst vornahm. Erst anschliessend stellte er die ersten Fragen:

1. Wann ist eine Zahlenfunktion berechenbar?
   ↴ ↴ Wenn es eine Registermaschine M gibt, die f berechnet. Ausserdem ist eine solche Funktion while-berechenbar. Weitere Antwort: eine berechenbare Funktion ist ?-rekursiv.

2. Wann ist eine Wortfunktion berechenbar?
   ↴ ↴ Wenn es eine Turing- oder Bandmaschine gibt, die f berechnet.

3. Sind die beiden äquivalent?
   ↴ ↴ Ja, sie haben die gleiche Berechnungsstärke (ohne Beweisführung).

4. Zusammenhang zwischen Zahlen- und Wortfunktionen?
   ↴ ↴ Standardnumerierung ?. Eine Wortfunktion g ist berechenbar, gdw. ihre Verschlüsselung ?^-1g?(Pfeil) eine berechenbare Zahlenfunktion ist.

   Exkurs: Churchsche These (zum Verständnis)

5. Was ist eigentlich eine Standardnumerierung?
   ↴ ↴ Keine Definition, allgemeine Erklärung aus den Kapiteln 7.1 und 8.1 geliefert, hat für den Augenblick wohl auch genügt.

6. Welche Programmiersprache, von den vielen, die es heute gibt, lernten wir im Kurs kennen?
   ↴ ↴ Das war die Modellprogrammiersprache ? ? (Diese Definition (? : N ↴ ↴ P(1)) war aufzuschreiben und verbal zu erklären)

7. Wie sehen denn die Programme dazu aus?
   ↴ ↴ Das sind die natürlichen Zahlen N.

8. Welche Anforderungen stellt man an eine Programmiersprache?
   ↴ ↴ Anforderungen (U) und (S) erläutert. Dazu die zugehörigen Theoreme utm und smn kurz erwähnt.

9. Sehr schön – definieren Sie mir doch bitte einmal das utm-Theorem.
   ↴ ↴ Definition Satz 8.5 (genau!)

10. Gibt es zu ? eine gleichwertige Programmiersprache?
    ↴ ↴ Äquivalenzsatz von Rogers (Satz 8.12) genau (!) definiert.

11 Kennen Sie den Satz von Rice?
    ↴ ↴ Hier musste ich mich kurz besinnen – nicht, weil ich den Satz nicht kannte, sondern weil ich bei einem Detail unsicher wurde. Das war kein Problem, ich fand Zeit zum Nachdenken, die nicht durch den Professor unterbrochen wurde.
    Ja, dann erfolgte die genaue (!) Definition des Satzes 9.17, die nähere Erklärung lieferte der Professor im Anschluss selbst.

12. Wir haben uns vorhin über Numerierungen unterhalten. Was machen wir denn bei reellen Zahlen?
    ↴ ↴ Wir benutzen die Darstellung reeller Zahlen durch schnell konvergierende (Geschwindigkeit 2^-1) Cauchy-Folgen rationaler Zahlen

13. Warum können wir keine Numerierung nehmen?
    ↴ ↴ Die Menge R ist nicht abzählbar.

14. Kurzer Einwurf von Prof. Weihrauch an dieser Stelle: Typ-2-Maschine. Was können Sie uns denn in diesem Zusammenhang dazu erzählen?
    ↴ ↴ Funktionsweise dieses Maschinentyps erklärt, danach, dass (und warum) die Maschine die Multiplikation von Dezimalbrüchen falsch berechnet.
    ↴ ↴ Der Rest zur Funktionsweise wurde mit dem Hintergrund zur Entwicklung dieses Modells vom Professor selbst erläutert.

15. Wie lautet der wichtigste Satz in Kurseinheit 7?
    ↴ Das ist der „Hauptsatz“ (Satz 10.20), nach dem jede (?,...,?)-berechenbare Funktion f : R^k ↴ R stetig ist.

Insgesamt kann ich mich meinen VorgängerInnen nur anschliessen: Prof. Weihrauch ist ein sehr angenehmer Prüfer, der sich der psychischen Verfassung seiner „Besucher“ sehr bewusst ist. Er trug durch seine ruhige und sehr freundliche Art, sowie dem Willen, die Prüfung möglichst als Gespräch durchzuführen und Pausen zum Nachdenken zu lassen, sehr zum Gelingen bei. Darüber hinaus sieht er die Prüfung als willkommene Gelegenheit, wissenswerte Dinge darzustellen, die nicht im Kurs stehen oder häufig nicht klar genug sind.

Man sollte aber nicht verkennen, dass Prof. Weihrauch trotz guter Laune sehr grossen Wert auf die genauen Definitionen und Sätze legt. Dann kommen, zumindest zu diesem Punkt, keine „nachbohrenden“ Fragen. Er schaut schon sehr genau, ob ein Prüfling den „roten Faden“ durch den Kurs und die Zusammenhänge erkennt. Sein Hauptaugenmerk liegt allgemein auf dem Teil A. Aus Zeitgründen wurde der Teil B bis auf die o. g. Einleitung nicht erwähnt.

Da meine Vorbereitung auch ein intensives Studium der Prüfungsprotokolle beinhaltete, bitte ich auch die Leser dieses Protokolls den „Generationenvertrag“ aufrecht zu halten. Eure Nachfolger werden es Euch danken.

Viel Glück, wenn Ihr „dran“ seit.

Vordiplom Theoretische Informatik
(Gedächtnisprotokoll)

Datum:	25.02.02
Prüfer:	Prof. Dr. Weihrauch
Beisitzer:	Herr Schröder
Dauer:	30 Minuten
Note:	2,3

Fragen zur Theoretischen Informatik A:

- Wann ist eine Zahlenfunktion f berechenbar?
  Wenn es eine Registermaschine M gibt mit $f = f_M$

- Wann ist eine Wortfunktion f berechenbar?
  Wenn es eine Turingmaschine M gibt mit $f = f_M$

- Zusammenhang zwischen Zahlenfunktion und Wortfunktion?
  Standardnumerierung ν: Eine Wortfunktion g ist berechenbar, wenn ihre „Verschlüsselung", die Zahlenfunktion $f = \nu^{-1} g \nu$ berechenbar ist.

- Ist die Standardnumerierung berechenbar?
  Nein, denn die Definition lautet ja ν : IN -> Σ^*.

- Rekursionssatz
  Satz 8.10

- Definition Numerierung
  Eine Numerierung einer Menge M ist eine (möglicherweise partielle) surjektive Funktion ν: IN -> M.

- Anforderungen (U) und (S) an vernünftige Programmiersprachen sowie Definitionen der utm- und smn-Theoreme.

- Satz über die Äquivalenz des smn-Theorems und der effektiven Komposition
  Es sei ψ: IN -> $P^{(1)}$ eine Numerierung von $P^{(1)}$, die das utm-Theorem erfüllt. Dann sind die folgenden Eigenschaften äquivalent:
  (1) smn
  (2) Es gibt r aus $R^{(1)}$ mit $\psi_{r(i,j)} = \psi_i \circ \psi_j$

- Was ist das Besondere an φ?
  Wesentlich ist, dass die Numerierung φ durch das utm - Theorem und das smn – Theorem bis auf Äquivalenz eindeutig bestimmt ist.
  (Äquivalenzsatz von Rogers)

- Ist das Komplement IN\A der rekursiven Menge A auch rekursiv?
  Ja! Weil man bei der charakteristischen Funktion cf_A nur die 1 und die 0 vertauschen muss und somit die $cf_{IN\setminus A}$ erhält.

- Kennen Sie eine Menge, die rekursiv-aufzählbar, aber nicht rekursiv ist?
  Die Menge K_φ

- Beweis, dass K_φ r.a., aber nicht rekursiv ist:
  Man muss zeigen, dass K_φ r.a. ist, das Komplement aber nicht.
  K_φ ist r.a. – das zeigt man über das utm – Theorem
  $IN \setminus K_\varphi$ ist nicht r.a. – das zeigt man über Diagonalisierung:
  Ich betrachte ein beliebiges $i \in IN$:
  $i \in IN \setminus K_\varphi \Leftrightarrow i \notin K\varphi \Leftrightarrow i \notin Def(\varphi_i)$
  also ist $IN \setminus K_\varphi \neq Def(\varphi_i)$ und kann damit, da ($Def(\varphi_0)$, $Def(\varphi_1)$,...)
  alle r.a. Mengen aufzählt, nicht rekursiv-aufzählbar sein.

- Wie ist die Berechenbarkeit auf anderen Mengen definiert?
  Berechenbarkeit auf beliebigen abzählbaren Mengen kann mittels Numerierung auf Berechenbarkeit auf IN zurückgeführt werden.

- Überabzählbare Mengen: Typ-2-Maschine, Erklärung, Besonderheiten, Darstellung

Fragen zur theoretischen Informatik B:

- Definition der regulären Mengen über rechtslineare Grammatik:
  Regeln der rechtslinearen Grammatik
  Regeln der rechtslinearen Normalformgrammatik
  Transformationen an Beispielen erklärt, wobei auf die Eliminierung längentreuer Regeln besonders eingegangen wurde (problematisch: A -> B, B -> C, C -> A)

- Kennen Sie eine kontextfreie Sprache, die nicht regulär ist?
  $L = \{ a^n b^n \mid n \in IN \}$

- Was sagen Sie zu
  $L = \{ wwR \mid w \in \{0,1\}^* \}$
  Nichtdeterministisch kontextfrei, dabei anschaulich erläutern, wie der Kellerautomat arbeitet und die Sprache erkennt.

Ich empfand die Prüfung als angenehm und locker. Prof. Weihrauch gab nach jeder richtigen Antwort ein positives Feedback, so dass die Nervosität relativ schnell verflog. Auch als ich 3 oder 4 Fragen gar nicht bzw. nur unzureichend beantworten konnte, war die Atmosphäre positiv und Prof. Weihrauch versuchte zu helfen, bzw. gab die Antwort selbst.
Die Note ist absolut gerecht.
Ich kann mich allen anderen nur anschliessen und Prof. Weihrauch uneingeschränkt empfehlen.

Alles Gute für Eure Prüfung!

Vordiplom Theoretische Informatik
Kurs 1653/ 1654 (Version WS 00/01 und SS 01)

Termin:	06.08.2001
Prüfer:	Prof. Dr. Weihrauch
Beisitzer:	Herr Schröder
Dauer:	20 Minuten
Note:	1,0

A Berechenbarkeit

- Wann ist eine Zahlenfunktion berechenbar?
  → Registermaschine

- Weitere Modelle?
  → WHILE Berechenbarkeit, μ-Rekursion

- Warum wird die Formulierung „heißt berechenbar“ benutzt?
  → Churchsche These

- Wann ist eine Wortfunktion berechenbar?
  → Turingmaschine, Bandmaschine

- Wie ist der Zusammenhang zwischen der Berechenbarkeit von Wort- und Zahlenfunktionen?
  → Standardnumerierung ν. Wortfunktion g ist berechenbar, gdw die Zahlenfunktion ν^(-1)gν berechenbar ist.

- Welche beiden Anforderungen stellt man an Programmiersprachen?
  → Anforderungen (U) und (S) verbal erläutert. Erwähnung des UTM und des SMN Theorems. Genaue Definition wurde nicht gefordert.

- Programmiersprache φ. Wie sehen hierbei die Programme aus?
  → natürliche Zahlen

- Wann sind zwei Programmiersprachen äquivalent?
  → Äquivalenzsatz von Rogers

- Wann ist eine Teilmenge A von N rekursiv?
  → charakteristische Funktion und wenn A=f^-1(0) für eine total berechenbare Funktion f.

- Wann ist eine Teilmenge A von N rekursiv aufzählbar?
  → Definitionsbereich einer partiell berechenbaren Funktion, Bildbereich einer total berechenbaren Funktion

- Gibt es noch eine weitere Charakterisierung rekursiv aufzählbarer Mengen?
  → Projektionssatz

- Exkurs in die Logik: Rekursivität der Menge der Axiome, die Menge der daraus ableitbaren Sätze ist rekursiv aufzählbar

- Wie sieht es mit der Berechenbarkeit auf anderen Mengen aus?
  → Numerierung, Notation, Darstellung

- Warum kann für die reellen Zahlen keine Numerierung existieren?
  → da die Menge der reellen Zahlen nicht mehr abzählbar ist

- Wie ist die Berechenbarkeit auf reellen Zahlen definiert?
  → ρ-Darstellung: Darstellung der reellen Zahlen durch schnell konvergierende Cauchy-Folgen rationaler Zahlen

- Warum kann man die reellen Zahlen nicht einfach durch unendliche Dezimalbrüche darstellen?
  → Multiplikation nicht berechenbar

- Ein Beispiel für eine berechenbare reelle Funktion?
  → f(x)= -x, f(x,y)= x*y

Fazit: Es war sehr hilfreich, daß ich Definitionen und Sätze (mehr oder weniger) exakt formulieren konnte. Ich glaube, daß ich mir damit einige weitergehende Fragen erspart habe und jedes Thema aus Teil A nur kurz angesprochen wurde. Wie auch schon andere in den Protokollen berichtet haben, führt Prof. Weihrauch während der Prüfung gerne noch weitere Details aus. Dann sollte man sich wirklich entspannt zurücklehnen! Obwohl ich zu dem Exkurs in die Logik außer gelegentlichem Kopfnicken und verständnisvollem Gemurmel nichts beitragen konnte, hat das der Note keinen Abbruch getan. Prof. Weihrauch hat allerdings zu Beginn des Exkurses auch erwähnt, daß er das mehr oder weniger als „Sternchenfrage“ betrachtet. Also: keine Panik!
Ich habe übrigens nicht vergessen, ein Protokoll für Teil B anzufertigen. Das wurde schlicht und ergreifend nicht geprüft!! Weder Komplexitätstheorie noch Formale Sprachen.

Viel Glück bei eurer Prüfung!

Prüfungsprotokoll Vordiplom Theoretische Informatik

30.3.2001
Prof. Weihrauch
Dauer: 30 min.
Note: 1,7

Teil A:

Wann heißt eine Zahlenfunktion berechenbar?
Registermaschine, WHILE-Programm, μ-Rekursion

"Heißt berechenbar", hat das einen realen Bezug?
Churchsche These

Wann heißt eine Wortfunktion berechenbar?
Turingmaschine

Zahlenberechenbar – wortberechenbar – gibt es da irgendeinen Zusammenhang?
Stichwort: Numerierung (hier wollte ich weit ausholen, Ordnungsfunktion, etc., aber Dr. Weihrauch hat mich gebremst, wollte bloß die Formel:)
$f = \nu^{-1}g\nu$, f Zahlenfunktion, g Wortfunktion, wenn f berechenbar, dann auch g.. ich durfte mich auf die einstelligen Funktionen beschränken.

Zahlenfunktionen, Wortfunktionen, Numerierungen, Programmiersprachen.... wir haben da versucht, ein wenig Ordnung reinzubringen...
Stichwort φ: ich habe erst die Definition hingeschrieben ($\varphi_i := \xi\nu_M\nu_P$) und dann kurz die einzelnen Teile erläutert: Numerierung der Bandprogramme, Bandmaschinen, partielle berechenbare Funktion, etc.., das hat ihm schon genügt.

Was sind hier die Programme?
die natürlichen Zahlen

Ist diese Numerierung so gut wie jede andere?
Äquivalenzsatz von Rogers erklärt

Welche Forderungen stellt man denn allgemein an Programmiersprachen?
U und S erklärt

Wie lautet das utm-Theorem?
Formel hingeschrieben und kurz erläutert

Und das smn-Theorem?
Hier mußte ich passen, mit dieser Frage hatte ich nicht gerechnet und konnte die Formel nicht mehr. Ich hab bloß irgendwas von wegen Komposition und Indizes gefaselt, es dann aber doch lieber sein gelassen...Prof. Weihrauch hat mich dann mit gespielter Strenge angesehen, dabei aber gelächelt.....

Gibt es überhaupt Funktionen, die nicht berechenbar sind?
Ja, klar (Einwurf von Prof. Weihrauch: Sonst bräuchten wir ja diesen ganzen

Berechenbarkeitskram nicht). Ich habe dann die Funktion
$g(x) := div$ falls $x \in Def(\varphi_i)$, 0 sonst aufgeschrieben. Beim Beweis bin ich ein wenig ins Schleudern geraten (Stichwort: Diagonalisierung, ich habe erst ewig um den heißen Brei rumgeredet, bis dieses Wort endlich fiel)

Wann ist eine Teilmenge von N rekursiv?
Charakteristische Funktion, Urbild der Null einer totalen berechenbaren Funktion.

Ist 3n rekursiv?
Ja.

Wie beweist man das z.B.?
Durch eine Registermaschine mit den beiden Ausgängen 1 (für $x \in A$) und Null (für $x \notin A$), aufzeichnen brauchte ich sie nicht (Prof. Weihrauch meinte, er traue es mir schon zu, daß ich so eine Registermaschine entwerfen könne).

Wann ist eine Teilmenge von N rekursiv-aufzählbar?
Es gibt eine totale, berechenbare Funktion f, für die gilt: A = Bild(f), Definitionsbereich der berechenbaren Funktionen.

Zusammenhang rekursiv/ rekursiv-aufzählbar?
Rekursive Mengen sind eine echte Teilmenge der rekursiv-aufzählbaren, r.-a. und Komplement r.-a. $\rightarrow$ Menge ist auch rekursiv

Eine Menge, die r.-a., aber nicht rekursiv ist?
Ich habe dann kurz $K\varphi$ erläutert. Als ich dann jedoch die korrekte mathematische Definition aufschreiben sollte, kam ich
ein wenig ins Schleudern (soooo genau wußte ich die nicht) und als er dann einen Beweis dafür wollte, daß $K\varphi$ rekursiv-aufzählbar ist, stand ich total auf der Leitung (er definierte dann die Funktion $h(i) = u_\varphi(i, i) = \varphi_i(i)$, die ist nach dem utm-Theorem berechenbar $\Rightarrow K\varphi$ ist r.-a.).

Teil B:

Zusammenhang s_M und t_M ?
$s_M(x) \leq t_M(x) + c$ (anschauliche Erklärung, warum)
$t_M(x) \leq c^{sM(lg(x)}$ falls $log \in s_M$ (auch hier anschauliche Erklärung:
Anzahl der Konfigurationen begrenzt, wenn der Speicher begrenzt ist.....
dann fragte er noch, wieso es denn exponentiell sei, da mußte ich leider passen)

Was heißt NP-vollständig?
Eine Menge X ist NP-vollständig, gdw. sie NP-hart ist und $X \in NP$.
Eine Menge ist NP-hart, wenn sich alle anderen NP-Mengen polynominell auf sie reduzieren lassen.

Kennen Sie NP-vollständige Mengen?
SAT-Problem, Clique, Traveling Salesman, Rucksack, Hamilton-Kreis....

Sagt Ihnen der Satz von Savitch etwas?
Satz aufgeschrieben und kurz erläutert.

Dann war – endlich – die Zeit vorbei.

Allgemeiner Eindruck: Ich war schrecklich nervös, es war meine letzte noch fehlende Vordiplomsprüfung und gilt allgemein als die Schwerste. Prof. Weihrauch nimmt einem aber schnell die Angst, man merkt, daß er nicht nur nach Lücken im Wissen sucht, sondern eher versucht einzuschätzen, was man denn nun eigentlich weiß. Ich bin doch einige Male ins Schleudern geraten bzw. total auf der Leitung gestanden und freue mich daher umso mehr über die 1,7. Er legt viel Wert darauf, daß man den Stoff verstanden hat und diesen auch gut wiedergeben kann. Die Definitionen sollte man detailiert beherrschen (hier habe ich ein wenig geschlampt), bei den Beweisen genügt im allgemeinen die Beweisidee. Teil B fiel bei mir sehr kurz aus – Fragen zu den Sprachen kamen garnicht – weil wir so lange bei Teil A geblieben sind. Ich kann Prof. Weihrauch als Prüfer nur empfehlen. Die Protokolle sind zur Vorbereitung super, ich hatte keine Frage, die nicht in dem einen oder anderen Protokoll schon mal vorkam. Wenn man diese Fragen alle beherrscht, dann dürfte eigentlich nichts schiefgehen. Viel Glück!!

Prüfer: Prof. Dr. Weihrauch
Datum: 31.10.2000
Dauer: ca. 25 Minuten
Note: 1,0

-Wann ist eine Zahlenfunktion berechenbar (Registermaschine, y-rekursiv. WHILE-Programme und anderes z.B. Markov)

-Unterschied Register/WHILE (Register GOTO)
-Zusamenhang Jede Regeisterfunktion durch eine WHILE-Schleife mit einem GOTO-Schritt

-f(x) := x^5 berechenbar (ja man kann Registermachine definieren)

-berechnbare Wortfunktion (Turingmachine,)

-Zusammenhang Turing/Band (mit Beweisidee)
-wie simuliert man bei Bandmaschine Kopfbewegung der Turingmaschine (Spuren, Bandmarkierung)
-wie erhöht sich der Zeitbedarf (quadratisch wegen der Simulation der Kopfbewegungen)

-Zusammenhang berechenbare Wortfunktion/Zahlenfunktion (g berechenbar <=> $v^{-1}gv$ berechnbar)

-Programmierpsrache (klein-vieh)
-Syntax (Nummer=Programme)
-Grundlegende Forderungen (utm, smn)
-satz utm-Theorem
-Äquivalenz smn <=> es gibt r: $klein_phi_{r(i, j)} = klein_phi_{(i)}$ 0 $klein_phi_{(j)}$
-Gibt es andere äquivalente Nummerierungen (Äquivalensatz von Rogers)

-rekrsiv und r.a Mengen
-Rekursiv (charakteristische Funktion)
-r.a. (M=Def(f) f partiuel, M=leere Menge oder M=Bild(f) F total)
-nicht rekursive Menge (K-vieh)
-Zusammenhang rekursiv/r.a (Rekursiv => r.a) (A rekursiv <=> A r.a und N\A r.a.)
-Beweis wie folgt aus (A r.a und N\A r.a) => A rekursiv (zwei Maschine, solange bis eine hält)
-welche Mengen sind nicht r.a (Äquivalenzproblem)
-Rekursionsatz
-Anwendung (Selbstreproduktionssatz, Beweis Satz von Rice)
-Wie lautet der Satz von Rice

-Berechenbarkeit von rationalen zahlen (Nummerierung)
-Berechenbarkeit von reelen Zahlen (nicht nummerierbar, Typ-2-Maschine mit Nachteilen, chauchy-Folgen)

-Zeigen Sie mir ein Verfahren mitttel desssen man einem aus endlichem Automaten reguläre Ausdrücke gewinnt (Prof. Weihrauch sagte, dies sei wirklich eine schwierige Frage und mir schien es, er wollte damit andeuten, daß ich dies nicht unbedingt wissen müssen. Ich hab die Formel aber tatsächlich auf die Reihe bekommen. gemeint ist das Verfahren aus dem Beweis von Lemma 8.3.15)

-0^n1^m regulär?/kontextfrei? Ja/ja
-0^n1^n regulär?/kontextfrei? Nein/ja (wenn n beschränkt wäre, dann regulär, weil Automat mit entsprechender Zustandszahl die Sprache erkennen kann, da n hier aber unbeschränkt ist, nicht regulär)

-Satz von Savitch
-P und NP (Definitionen)
-Was ist NP-Vollständigkeit

Prüfungsprotokoll Diplomvorprüfung Theoretische Informatik A und B

Juli 2000
Prüfer: Prof. Weihrauch
Dauer: 30 Minuten

1. Berechenbarkeit von Zahlenfunktionen.
   "Hat die Registerberechenbarkeit eigentlich irgendetwas mit Computern zu tun?" (Churche These)
2. Berechenbarkeit von Wortfunktionen.
3. Zusammenhang von Wort- und Zahlenberechenbarkeit.
4. Bandmaschinen, informale Beschreibung, Bandalphabete.
   "Es gibt eine bestimmte normierte Bandmaschine." (Ohne Hilfssymbole)
   "Wie macht man das, dass man ohne Hilfssymbole auskommt?" (Beweis zu Satz 6.6)
5. Ny,Ny-Berechenbarkeit.
6. Berechenbarkeit reeller Funktionen.
   Rho-Darstellung reeller Zahlen erklären. Exakte Definition wurde nicht verlangt. "Welche besondere Eigenschaft muss die Cauchy-Folge haben?" (schnell konvergierend, mit Geschwindigkeit 2ˆ-k)
7. Rekursive und rekursiv aufzählbare Mengen.
   "Gibt es rekursiv aufzählbare Mengen, die auch rekursiv sind?" (Wenn das Komplement auch rekursiv aufzählbar ist) Beweisidee für: A rekursiv => A rekursiv aufzählbar. "Gibt es eine Menge die rekursiv aufzählbar aber nicht rekursiv ist ?" (K_phi,mit Definition)
   "Was bedeutet das phi?"(Definition und Herleitung über Bandprogramme)
8. Die Forderungen U und S an Programmiersprachen erklären. Definition des utm-Theorems. "Wann ist eine Nummerierung äquivalent zu phi?" (Äquivalenzsatz von Rogers)
9. Rice Theorem.
10. Reguläre Sprachen.
    Konstruktion eines regulären Ausdruckes aus einem endlichen Automaten.
11. Kontrollturingmaschine. Beschreibung .Erkannte Sprache.
12. Definitionen von: NP, NP-vollständig, NP-hart.
13. Beispiel eines NP-vollständigen Problems. (3-SAT)

Teilweise wurden Fragen gestellt, bei denen ich den Eindruck hatte, dass sie etwas über den geprüften Kurstext hinausgingen Die Antworten hat Prof. Weihrauch dann selbst ausführlich gegeben. Der Zweck ist wahrscheinlich der, dass man die Zusammenhänge noch besser erkennt. In die Beurteilung sind diese Fragen jedenfalls nicht eingegangen, daher sollte man sich dadurch nicht nervös machen lassen.
Die Prüfungsprotokolle der letzten 3 Jahre haben sich zur Vorbereitung sehr gut geeignet.

Diplom Vorprüfung

Einführung in die Theoretische Info, 1653 (10/96) und 1654 (4/97).
Prüfer: Prof. Weihrauch
Datum: 13.6.2000
Dauer: ca. 25 min.

1) Definition für eine berechenbare Zahlenfunktion?
   ⇒ Über Registermaschinen
2) Andere Definition?
   ⇒ Über μ–Rekursion. μ–rekursive Funktionen sind genau die Register-berechenbaren.
3) Was ist der Unterschied zwischen „ist berechenbar“ und „heißt berechenbar“ ?
   ⇒ „Heißt berechenbar“ bedeutet, die mathematische Definition (s.o.) ist erfüllt. Daß die Funktion dann auf dem Papier oder mit einem Rechner „berechenbar ist“ besagt die Churchsche These.
4) Definition für berechenbare Wortfunktionen?
   ⇒ Es gibt eine Bandmaschine, die die Wortfunktion berechnet.
5) Zusammenhang zwischen berechenbaren Zahlen- und Wortfunktionen?
   ⇒ Meine Antwort: Sei f eine Wortfunktion. f ist genau dann berechenbar, wenn es eine Numerierung ν gibt, mit g:= $\nu^{-1} f \nu$, so daß g (= Zahlenfunktion) berechenbar ist.

   Kann das eine beliebige Numerierung oder muß es eine Standardnumerierung sein?
   ⇒ Meine Antwort: Numerierung kann beliebig sein. Das war leider falsch. Es muß schon eine Standardnumerierung sein. (Problem ist wahrscheinlich, daß es auf keinen Fall eine Numerierung sein darf, die selber nicht berechenbar ist. Sonst sieht es mit der Berechenbarkeit von $\nu^{-1} f \nu$ natürlich schlecht aus...)
6) Definition für rekursive Mengen?
   ⇒ $\exists\, f \in P:\ A = f^{-1}\{0\}$
7) Ist die Menge A:={1, 10, 100, 1000, ...} rekursiv?
   ⇒ Ja, denn f(n):= 10^n ist rekursiv.

   Das reicht aber noch nicht. Wie sieht die Überprüfung für eine Zahl z konkret aus?
   ⇒ Ich berechne für die Zahlen 0, 1, 2, ... die Funktion f(n) und vergleiche das Ergebnis mit z. Wenn die Differenz nicht Null wird, bis f(n) > z ist, dann ist z nicht aus A. Sonst schon.

   Man kann also aufhören, wenn f(n) > z. Man nutzt hier also die Monotonie von f aus. Wäre f nicht monoton, würde das so noch funktionieren.
8) Definition für rekursiv aufzählbar?
   ⇒ A ist r.a. gdw. $\exists\, f \in P^{(1)} : A = Def(f)$

   Andere Definition?
   ⇒ A = {} oder $\exists\, g \in R^{(1)} : A = Bild(g)$
9) Abgeschlossenheit für rekursive Mengen?
   ⇒ Für $A \cup B,\ A \cap B$ *und* $\overline{A}$.

   Beweis für $A \cap B$?
   ⇒ Sei A=$f^{-1}\{0\}$ und B=$g^{-1}\{0\}$. Dann ist $A \cap B = (f+g)^{-1}\{0\}$.

   Abgeschlossenheit für rekrusiv aufzählbare Mengen?
   ⇒ Für $A \cup B,\ A \cap B$. Aber nicht für das Komplement.
10) Beispiel für eine r.a. Menge, die nicht rekursiv ist?
   ⇒ K_φ . Dabei auch Definition von φ erklären...

11) Beispiel für eine Funktion, die nicht berechenbar ist?

⇒ g(i) = {div, falls i aus Def(φ_i). 0 sonst.
g ist ungleich jedem φ_i, (diagonal definiert...) deshalb ist g nicht aus $P^{(1)}$.

12) Definition für reguläre Mengen ?

⇒ a) wird von endl. Automaten erkannt. b) Es gibt eine rechtslineare Grammatik...
c)Definition als Abschluß mit den Operatoren $\cup$ $\cdot$ *und* $*$. (siehe B 2.1.2)

13) Beispiel für eine Menge, die nicht regulär ist?

⇒ $A = \{a^n b^n\}$

Warum ist diese Menge nicht regulär? (Anschauliche Argumentation...)

⇒ Weil ein endl. Automat (im Gegensatz zu einem Kellerautomaten) keinen Speicher hat, um sich zu merken, wie viele a's kamen, um dann die Zahl der b's zu überprüfen. Ein Kellerautomat kann's, A ist daher nicht regulär aber kontextfrei.

Wie lautet die kontextfreie Grammatik für A ?

⇒ $S \rightarrow aSb \mid \varepsilon$

14) Beispiel für eine Menge, die nicht kontextfrei ist?

⇒ z.B. die Menge der Pascal-Programme, weil jede Variable deklariert und dann verwendet werden kann. Es läuft darauf hinaus, daß B = {w$w} nicht kontextfrei ist.

Wie zeigt man das?

⇒ Mit dem Pumping-Lemma für kontextfreie Mengen.

15) Komplexitätstheorie: Kennen Sie einen Zusammenhang zwischen s_M und t_M ?

⇒ a) $s_M \leq t_M$ b) falls s_M mindestens logarithmisch wächst: $t_M \leq c^{SM} + c$.

Haben Sie eine anschauliche Erklärung dafür?

⇒ a) ist klar. Um eine Speichereinheit zu belegen ist immer eine Zeiteinheit nötig. Es können aber auch Schritte erfolgen, die nur Zeit, aber keinen Speicher kosten.
zu b) konnte ich keine Erklärung liefern. Von Prof. Weihrauch kam folgende Erklärung: Wenn der Speicher beschränkt ist, gibt es nur eine beschränkte Anzahl von Konfigurationen. Wenn die alle durchlaufen sind, muß die Maschine halten, sonst hat man eine Endlosschleife. Diese Überlegung liefert eine obere Schranke für die Zeit.

16) Definition für P, NP ?

⇒ Siehe Kurstext. Außerdem Erklärung, warum diese Definition in der Praxis relevant ist (Wenn nicht in polynomialer Zeit berechenbar, dann auch in der Praxis für große Zahlen nicht berechenbar, weil es zu lange dauern würde)

17) Definition NP vollständig, P-NP-Problem

⇒ siehe Kurstext.

Fazit: Herr Prof. Weihrauch ist ein sehr netter und wohlwollender Prüfer. Die Protokolle sind für die Vorbereitung super. Ich fand nicht, daß ich viele Definitionen oder Sätze sehr exakt (schriftlich) formulieren mußte. Das meiste reichte mündlich in Stichpunkten, oft fragte er nach einer Anschauung zu den Aussagen / Definitionen.
Mit meiner Note (1,0) bin ich natürlich sehr zufrieden, ich hätte nach der Prüfung eher eine 1,7 oder 1,3 erwartet, weil ich schon ein paar kleine Fehler gemacht habe (s.o.).

Diplom-Vorprüfung
(Gedächtnisprotokoll)

Prüfungsgebiet:	1653 Theoret. Inf. A (WS 97/98)
	1654 Theoret. Inf. B (SS 98)
Prüfer:	Prof. Weihrauch
Beisitzer:	Herr Schröder
Datum:	17.02.2000
Dauer:	ca. 30 Minuten

<u>Teil A</u>

Wann ist eine Zahlenfunktion f berechenbar?
⇨ Def. 3.2 (1), Satz 5.12 (3), Satz 7.17

Wann ist eine Wortfunktion f berechenbar?
⇨ Def. 6.2, Satz 6.5

Zusammenhang Turing- und Bandmaschine?
⇨ Beweise zu Satz 6.5 in groben Zügen dargelegt

Zusammenhang zwischen berechenbarer Zahlen- und Wortfunktion?
⇨ Standardnummerierung (Def. 7.1, Satz 7.6)

Berechenbarkeitskonzepte bei rationalen und reellen Zahlen?
⇨ Erläuterung der Begriffe Nummerierung, Notation, Darstellung und Erklärung, wann man was benutzt, dabei die Dezimaldarstellung (Beispiel 10.4.2) erwähnt und Typ-2-Maschine erklärt

Zusammenhang von Funktionen und Programmen?
⇨ Standardnummerierung φ von $P^{(1)}$ (ohne mathematische Details) erläutert.

Welche zwei wesentlichen Anforderungen stellt man an Programmiersprachen?
⇨ (U) und (S) erklärt (KE 5 Seite 1 und 2) und erwähnt, dass das utm- und das smn-Theorem mathematische Präzisierungen dieser Eigenschaften sind

Was ist denn bei φ die Syntax?
⇨ IN (vgl. Seite 1 KE 5)

Was ist mit anderen Nummerierungen? Wann sind sie äquivalent zu φ?
⇨ Satz 8.12

Was besagt der Rekursionssatz?
⇨ Rekursionssatz von Kleene (Satz 8.10) erläutert

Kennen Sie eine Anwendung dieses Satzes?
⇨ Satz 8.11

Wann ist eine Menge $A \subseteq IN$ rekursiv, wann r.a.?
⇨ Def. 9.1, Satz 9.2, Def. 9.8 genügten bereits

Was bedeutet denn „rekursiv“ und „rekursiv-aufzählbar“?
⇨ vgl. Seiten 1 und 6 der KE 6

Wie hängen die beiden Begriffe zusammen?
⇨ A rekursiv => A r.a. und Satz 9.11

Kennen Sie Mengen, die r.a., aber nicht rekursiv sind?
⇨ Satz 9.14

Teil B

Wie sehen die Regeln einer rechtslinearen Grammatik und ihrer Normalform aus?
⇨ Abschnitt 8.2 Seite 25 KE 5

Wie erhält man die Normalform?
⇨ Stichwort „Transformationen“ genügte schon, zu genaueren Erklärungen kam ich nicht mehr

Wie erhält man zu der von einem endlichen Automaten akzeptierten Sprache einen entsprechenden regulären Ausdruck?
⇨ über die Rekursionsformel im Beweis zu Lemma 8.8 (genaue Angabe sowie Erklärung der Formel)

Wie sind P und NP definiert?
⇨ Def. 3.5 und 4.2 sowie anschauliche (verbale) Beschreibung

Was weiß man über den Zusammenhang P – NP?
⇨ $P \subseteq NP$, aber ob P = NP gilt, ist bislang noch nicht geklärt und eines der zentralen Probleme der theoretischen Informatik.

deterministische Abschätzung von NP?
⇨ Korollar 4.2 (2)

Wie ist die von einer Kontrollturingmaschine (KTM) erkannte Sprache definiert?
⇨ vgl. Def. 4.1; dabei sprachen wir auch über den Aufbau und die Funktionsweise der KTM, insbesondere legte ich auch dar, was τ_M (x,y) = 1 bzw. τ_M (x,y) = 2 bedeutet (vgl. einleitende Erklärungen vor Def. 4.1)

Fazit: Die Prüfung verlief in ruhiger und sehr freundlicher Atmosphäre. Prof. Weihrauch leitete jede Frage mit ein paar Ausführungen ein. Bisweilen verlief die Prüfung eher in Dialogform statt als Frage-Antwort-Spiel. Ich hatte immer genug Zeit zum Nachdenken und konnte auch mal zurückfragen, wenn ich nicht gleich wusste, worauf Prof. Weihrauch hinaus wollte. Über die Benotung (1.0) habe ich mich natürlich sehr gefreut. Insgesamt kann ich Prof. Weihrauch als Prüfer nur empfehlen!

Viel Glück bei eurer Prüfung und vergesst das Prüfungsprotokoll nicht!

Gedächtnisprotokoll: Vordiplom Theoretische Informatik A/B
1653 (10/98), 1654 (4/99)
Prof.Dr.K. Weihrauch (14.02.2000)
Dauer: ca. 20 - 25 min.

Nach einleitenden, einstimmenden Worten:

- Wann heißt eine Zahlenfunktion brechenbar?
  Wenn sie von einer Registermaschine berechnet werden kann.

- Wann heißt eine Wortfunktion berechenbar?
  Wenn es eine Turing- oder eine Bandmaschine gibt, die diese berechnet.

- Zusammenhang zwischen Turingmaschine mit vielen Bändern und Bandmaschine mit einem Band. Warum können gleiche Funktionen berechnet werden.
  Da es zu jeder Turingmaschine eine Bandmaschine gibt, die dieselbe Funktion berechnet.
  Beweisidee zu Simulation der Turingmaschine durch die Bandmaschine
  Zusammenkleben der Bänder, ein Band entspricht einer Spur der Bandmaschine mit den neuen Symbolen, die Tupel sind. Außerdem nötig ist je Symbol des Arbeitsalphabets der Turingmaschine ein überstrichenes Symbol zur Kennzeichnung der Kopfstellung je Spur.
  Simulation der Befehle. (wie in KE)

- Zeitbedarf im Vergleich zur Turingmaschine?
  Zu beachten, jeweils zusätzlich das überstrichene Symbol, die Kopfstellung ist zu suchen. (quadratisch)

- Zusammenhang zwischen Zahlenfunktionen und Wortfunktionen?
  gegeben durch Standardnumerierung, also einer Numerierung bzgl. einer Ordnungsbijektion. (s. KE)

- Außer Zahlen- und Wortfunktionen auch Funktionen numerierbar.
  Hierzu definierte Programmiersprache?
  klein-vieh, die Standardnumerierung der einstelligen partiellen, berechenbaren Zahlenfunktionen, mit natürlichen Zahlen als Programmen.

- 2 fundamentale Anforderungen an Programmiersprachen hierzu:
  (U) die universelle Maschine, z.B. ein Digitalrechner, der mit Eingabe eines beliebigen Programmes und e i n e r Eingabe für dieses Programm einen Funktionswert berechnet,
  (S) Übersetzung, daß z.B. aus 2 Programmen mit 2 Indizes bzgl. klein-vieh ein Index für die Komposition dieser beiden Programme als Funktionswert der totalen einstelligen Funktion r berechnet wird.
  (U) und (S) sind nötig für effektives Programmieren.

Definieren des utm-Theorems, des smn-Theorems (wie im Kurstext).
- Ein zentraler Punkt, der besagt, daß es bis auf Äquivalenz nur eine sinnvolle Programmiersprache gibt:
Siehe Äquivalenzsatz von Rogers, der dies beinhaltet.
Definition des Satzes wie im Kurstext.

-Wann ist eine Teilmenge von N rekursiv: Wenn die charakteristische Funktion berechenbar ist.
-Wann rekursiv aufzählbar: Wenn sie der Definitionsbereich einer einstelligen berechenbaren partiellen Funktion ist.
Eine Menge, die r.a. aber nicht rekursiv ist?
K-vieh, das Selbstanwendbarkeitsproblem.
Definition, wie im Kurstext (angeschrieben).
-Wie zeigt man, daß K-vieh r.a.:
K-vieh ist der Def.bereich eines h(i) = u-vieh(i,i).
-Wie hängen rekursiv und r.a. zusammen?
Eine rekursive Menge ist aufjeden Fall r.a.
Ist eine Teilmenge aus N und ihr Komplement bzgl. N r.a. dann ist sie auch rekursiv.

Überleitung zu Teil B.
- Regeln der rechtslinearen Grammatik: A-> wB und A-> w
der rechtslinearen Normalformgrammatik: A-> aB und A-, Epsilon
(dabei A,B je em Nichtterminal, w ein Terminalwort, a ein Terminalsymbol)
-Wie transformiert man eine rechtslineare- in eine rechtslineare Normalformgrammatik?
Durch Normieren der Terminalregeln,
Verkürzen der Regellängen,
Eliminieren längentreuer Regeln (s. Kurstext)
-Welche Form kennt man zur Syntaxdefinition von Pascal?
Backus Naur Form.
Ist sie kontextfrei? Ja.
Ist Pascal kontextfrei? Nein, wegen der zusätzlichen Vorschrift, daß jede Variable vor ihrer Verwendung deklariert sein muß.

Zum Abschluß zur Komplexitätstheorie:
-Was ist P - NP?
P die Menge aller in polynomieller Zeitkomplexität entscheidbaren Sprachen.
NP die Menge aller in pol. Zeitkompl. beweisbaren Sprachen.
Problematik hierzu? P enthalten in NP, Gleichheit der Mengen nicht erwiesen, aber bislang nicht ausgeschlossen.
-Wie ist die von einer Kontroll-Turingmaschine erkannte Sprache definiert?
Definition wie im Kurstext (aufgeschrieben).
Bis auf diese Definition waren fast alle Antworten verbal zu geben.
Herr Prof. Dr. Weihrauch ist als Prüfer zu empfehlen, die Benotung ist wohlwollend,

die Art der Prüfung entspannt. Allerdings ist es nötig, die Zusammenhänge parat zu haben, grundlegende Definitionen exakt wiedergeben zu können, und kleine Anwendungen wie die Flußdiagramme zu charakteristischen Funktionen verbal darlegen zu können.

Viel Erfolg bei Deiner Prüfung.

Gedächtnisprotokoll:

Vordiplom Theoretische Informatik am 17.12.1999

Prüfer: Prof. Dr. Weihrauch
Prüfungsinhalt: Kurse 01653 und 01654
Dauer: 25 – 30 Minuten
Note: 1.7

Wann heißt eine Zahlenfunktion berechenbar ?
Wann ist eine Funktion y-rekursiv ?
Hilfssymbol-Lemma + Beweisidee
Wieso wählt man ein m fester Länge ?
Wenn man bei Bandmaschinen 1 Schritt nach links macht, was muß man jetzt machen ?
Ist f(x) := { 0 falls x Quadratzahl, div. sonst berechenbar ?
Wie zeigt man das ?
Was ist φ ?
Genaue Definition des utm-Theorems
Zusammenhang zwischen berechenbaren Wort- und Zahlenfunktionen
Wann heißt eine Teilmenge von IN rekursiv ?
Ist die Menge der natürlichen Zahlen rekursiv ?
Wenn eine Menge rekursiv aufzählbar ist, ist sie dann auch rekursiv ?
Gibt es denn eine Menge die rekursiv aufzählbar ist und nicht rekursiv ? ($K\varphi$)
Beweis, $K\varphi$ ist rekursiv aufzählbar (über utm-Theorem)
Wann ist eine Teilmenge u-rekursiv
Regelmenge der rechtslinearen Grammatiken
Regelmenge der rechtslinearen Normalformgrammatiken
Transformation von rechtslin. G in Normalformgrammatiken
Womit erkennt man kontextfreie Sprachen
Ist L1 : { $\omega\omega^R$. ...} regulär ? kontextfrei ? (Nein/Ja)
Ist L2 : { $\omega\omega$ . . } regulär ? kontextfrei ? (Nein/Nein Pumping Lemma)
Wie erkennt Kellerautomat, daß L1 kontextfrei ist ?
Woher weiß der Kellerautomat, ab wann er vergleichen muß
(der Kellerautomat ist nichtdeterministisch, Übergang ist jederzeit möglich)
A ε BAND(n^2) Was bedeutet das ?
B ε BAND(n^2) Ist B rekursiv ?
Sagt Ihnen der Satz von Savitch etwas ?

Gesamteindruck:
Prof. Weihrauch ist ein angenehmer Prüfer, der seine Fragen ruhig und ausführlich formuliert. Manchmal weiß man nicht, ob man sein Wissen schon bei der langen Fragestellung einfließen lassen soll. Ich habe es meistens nicht getan. Bei der Definition des utm-Theorems musste Prof. Weihrauch etwas nachhelfen. Bei L1 : { $\omega\omega^R$ . . . } hatte ich etwas Probleme und den Satz von Savitch konnte ich gar nicht. (Das war auch das Einzige, was in keinem alten Prüfungsprotokoll zu finden war.) Mit der Note bin ich sehr zufrieden. KE7 Information und Codierung aus Teil B prüft Prof. Weihrauch wohl generell nicht.
Mir haben die Prüfungsprotokolle sehr geholfen.

Viel Glück

Diplom-Vorprüfung
(Gedächtnisprotokoll)

Fachprüfung:	Einführung in die Theoretische Informatik
Prüfungsgebiet	1653 Theoretische Informatik A (10/98)
	1654 Theoretische Informatik B (04/99)
Prüfer:	Prof. Weihrauch
Datum:	29.07.1999
Dauer:	25 Minuten

- Wann ist eine Zahlenfunktion f berechenbar?
  $\Rightarrow$ Wenn es eine Registermaschine M gibt mit $f_M = f$.

- Wann ist eine Wortfunktion berechenbar?
  $\Rightarrow$ Wenn es eine Turingmaschine M gibt mit $f_M = f$.

- Sind Registermaschinen reale Objekte?
  Nein. Sie sind nur gedankliche Objekte.

- Wie ist der Zusammenhang zwischen den Zahlen- und den Wortfunktionen?
  $\Rightarrow$ Zusammenhang über die Standardnumerierung ν. Sei $f :\subseteq N \rightarrow N$. f berechenbar $\Leftrightarrow$ $\nu f \nu^{-1} :\subseteq \Sigma^* \rightarrow \Sigma$ berechenbar.

- Was meinte Herr Church mit seiner These?
  $\Rightarrow$ Die Register-berechenbaren Funktionen sind genau die maschinell berechenbaren Funktionen. Es gibt keinen Beweis dafür, aber die These wird als richtig angesehen.

- Herrn Turings Maschine hatte nur 1 Band, wir haben mehrere Bänder. Wie kommen wir zum gleichen Berechenbarkeitskonzept?
  $\Rightarrow$ Für jede Turingmaschine M mit k Bändern gibt es eine Bandmaschine N mit nur 1 Band, so daß $f_M = f_N$ und $t_M = t_N$.. Weitere Erklärung wie im Kurstext Kapitel 6.3.

- Als Vorgriff schon einmal: Welche Zeitkomplexität hat wohl die Bandmaschine?
  $\Rightarrow$ Im Zweifelsfall muß der Kopf um n Zeichen bewegt werden, und das für jedes Zeichen auf den Bändern. Die Zeitkomplexität ist also O (n^2).

- Wann ist eine Menge rekursiv?
  $\Rightarrow A \subseteq N^k$ ist rekursiv $\Leftrightarrow$ die charakteristische Funktion cf_A ist berechenbar:

$$cf_A(x) := \begin{cases} 1, \text{ falls } x \in A \\ 0, \text{ sonst} \end{cases}$$

  $A \subseteq N^k$ ist rekursiv $\Leftrightarrow$ es gibt eine Funktion $f \in R^{(1)}$ mit $A = f^{-1}\{0\}$.

- Wann ist eine Menge rekursiv aufzählbar?
  $\Rightarrow A \subseteq N^k$ ist rekursiv aufzählbar $\Leftrightarrow$ es gibt eine Funktion $f \in P^{(1)} : N^k \rightarrow N$ mit $A = Def(f)$.

- Wann besteht Abgeschlossenheit für diese beiden Mengen?
  $\Rightarrow$ Rekursive Mengen sind unter Vereinigung, Durchchnitt und Komplement abgeschlossen. Rekursive Mengen sind nur unter Vereinigung und Durchschnitt abgeschlossen.

- Was haben Sie für eine Beweisidee für $A_1 \cup A_2$ und $A_1 \cap A_2$ für rekursive Mengen?
  $\Rightarrow$ Zwei Registermaschinen M_1 und M_2 berechnen die Mengen A_1 und A_2. Für Vereinigung rechnen beide Maschinen abwechselnd. Wenn eine davon hält, gilt $A_1 \cup A_2$. Für den Durchschnitt rechnet M_1 mit dem Ergebnis von M_2 weiter. Wenn M_2 hält, gilt $A_1 \cap A_2$.

- Was heißt (ν, ν)-berechenbar?
  $\Rightarrow$ Sei $\nu :\subseteq N \rightarrow M_0$, M_1 eine Numerierung. Eine Funktion $f :\subseteq M_0 \rightarrow M_1$ heißt (ν, ν)-berechenbar $\Leftrightarrow$ es gibt eine berechenbare Funktion $g :\subseteq N \rightarrow N$ mit $f(\nu_1) = \nu_0 \circ g(i_1)$.

- Wenn wir andere Menge (z. B. rationale Zahlen oder irgendwelche Wörter aus einem Alphabet) berechnen wollen, wie lautet dann der Berechenbarkeitsbegriff?
  $\Rightarrow$

$$\begin{tikzcd} N \arrow[r, "g"] \arrow[d, "\nu_1"'] & N \arrow[d, "\nu_2"] \\ M \arrow[r, "f"'] & M \end{tikzcd} \qquad \begin{tikzcd} i \arrow[r, "g"] \arrow[d, mapsto, "\nu_1"'] & g(i) \arrow[d, mapsto, "\nu_2"] \\ x \arrow[r, "f"'] & f(x) \end{tikzcd}$$

  Wenn $\nu_1(i) = x \in Def(f)$ gilt, dann gilt $f(x) = \nu_2\, g(i)$: $f \circ \nu_1(i) = \nu_2 \circ g(i)$ für alle i mit $\nu_1(i) \in Def(f)$.

- Es sei $L = \{ ww^R \mid w \in \{0,1\}^* \}$. Ist diese Sprache regulär?
  $\Rightarrow$ Nein.

- Wie beweist man das?
  $\Rightarrow$ Mit einem nichtdeterministischen Kellerautomaten.

- Wie lauten die Regeln für diese Sprache?
  $\Rightarrow S \rightarrow 0\ S\ 0 \mid 1\ S\ 1 \mid \varepsilon$

- Ein nichtdeterministischer Automat ist keine reale Maschine. Wie kann man daraus eine reale Maschine konstruieren?
  $\Rightarrow$ Zu jedem endlichen (nichtdeterminierten) Automaten A gibt es einen determinierten endlichen Automaten B mit $L(A) = L(B)$.

- Wie geht man dabei vor?
  $\Rightarrow$ Man muß einen Potenzautomaten konstruieren, indem man jeden möglichen Pfad verfolgt und dafür neue Zustände anlegt.

- Wie ist der Zusammenhang zwischen einem nichtdeterministischen Automaten und einem Potenzautomaten?
  $\Rightarrow$ Sei $A = (\Sigma, Q, q_0, F, \delta)$ ein nichtdeterminierter endlicher Automat. Dann ist ein vollständiger, determinierter Automat B wie folgt definiert: $B = (\Sigma, Q', q'_0, F', \delta')$ mit
  $Q' := 2^Q$ ist die Zustandsmenge von B,
  $F' := \{ P \subseteq Q \mid P \cap F = \emptyset \}$ ist die Endzustandsmenge,
  $q'_0 := \{ q_0 \}$ ist der Anfangszustand.

- Wie funktioniert eine Kontrollturingmaschine?
  $\Rightarrow$ Für ein beweisbares Problem benötigt man eine Kontrollturingmaschine. Sie hat neben dem üblichen Eingabeband und den Arbeitsbändern eine Hilfseingabe auf Band 0, das ein Einweg-Leseband ist. Das Band 0 enthält einen „Beweis“ für die Eingabe auf Band 1. Die Maschine hält entweder mit einem akzeptierenden oder einem verwerfenden Zustand.

- Wie ist die genaue Definition der von einer Kontrollturingmaschine erkannten Sprache?
  $\Rightarrow$ Die von einer Kontrollturingmaschine M erkannte Sprache $L_M \subseteq \Sigma^*$ ist definiert durch: $L_M = \{ x \in \Sigma^* \mid \exists y \in \{0,1\}^*. \iota_M(x, y) = 1\}$. Das y stellt den nichtdeterministischen Anteil dar, den Beweis auf der Hilfseingabe.

Die sehr präzise gestellten Fragen orientierten sich weitgehend am Verlauf des Kurstextes. Die tatsächliche Reihenfolge war möglicherweise geringfügig anders als hier aufgeschrieben. Abgesehen von dem „Vorgriff“ bei der Bandmaschine wurden keine Fragen zur Komplexitätstheorie gestellt. Prof. Weihrauch legt größten Wert auf die exakte Wiedergabe der Definitionen und der Sätze. Eine Kenntnis der Zusammenhänge ist zwar unbedingt notwendig, aber ohne die Sätze und Definitionen geht gar nichts. Also: Sorgfältig lernen!

Ich wünsche viel Erfolg bei der Prüfung. Und vergeßt das Protokoll nicht!

Gedächtnisprotokoll zur Vordiplomprüfung in Theoretischer Informatik

Datum: 21.06.1999
Prüfer: Prof. Dr. Weihrauch
Beisitzer: Herr Schröder
Dauer: 30min
Kursversion: Überarbeitete Version, WS 97/98, SS 98

Fragen zum Teil A

- Wann heißt eine Zahlenfunktion berechenbar?
  Es gibt eine RM, die die Funktion berechnet, oder wenn die Funktion μ-rekursiv ist.

- Unterschied zwischen „heißt“ berechenbar und „ist“ berechenbar.
  Churchsche These

- Wann heißt eine Wortfunktion berechenbar?
  Es gibt eine Bandmaschine oder eine Turingmaschine, die die Funktion berechnet.

- Zusammenhang zwischen Wortberechenbarkeit und Registerberechenbarkeit.
  $f :\subseteq \Sigma^* \to \Sigma^*$ berechnbar $\Leftrightarrow \nu^{-1} f \nu :\subseteq N \to N$ berechenbar mit ν Standardnumerierung von Σ^*

- Besondere Numerierung von Programmen?
  Standardnumerierung φ erklärt

- utm- und smn- Theorem als besondere Eigenschaft. Bitte definieren Sie das utm-Theorem.
  Definition vgl. Kurstext

- Wozu braucht man das utm-Theorem?
  inhaltl. Erläuterung, vgl. Kurstext

- Rekursive Mengen.. .
  Definition über charakteristische Funktion

- Gibt es noch eine zweite Definition?
  A = $f^{-1}\{0\}$

- A, B rekursive Mengen, ist dann $A \cap B$ auch rekursiv?
  Ja! Mit Beweis, vgl. Kurstext

- Rekursiv-aufzählbare Mengen?
  Definition A r.-a., wenn es ein $f \in P$ gibt miz A = Def(P)

- A, B r.-a. Mengen, ist $A \cup B$ r.-a.?
  Ja! Mit Beweisidee, vgl. Kurtext

- Kennen Sie eine Menge, die r.-a., aber nicht rekursiv ist?
  $K_\varphi = \{i | \varphi_i(i)$ ex.$\}$

- Beweis, daß K_φ r.-a. ist
  Für h(i) = $u_\varphi(i, i)$ gilt Def(h)$=K_\varphi$

- Beweis, daß K_φ nicht rekursiv ist
  Beweis über direkte Diagonalisierung; vgl. Selbsttestaufgabe im Kurstext

Fragen zum Teil B

- Wann ist eine Sprache regulär? Suchen Sie sich eine aus den vielen Definitionen aus!
  Es gibt einen endl. Automaten, der die Sprache erkennt.

- Prof. Weihrauch schrieb fünf Sprachen auf, zu denen ich etwas sagen sollte:
  $L_1 = \{a^n b^m | n, m \in N\}$
  $L_2 = \{a^n b^n | n \in N\}$
  $L_3 = \{w\$w^R | w \in \Sigma^*\}$
  $L_4 = \{ww^R | w \in \Sigma^*\}$
  $L_5 = \{w\$w | w \in \Sigma^*\}$
  Zu 1: L ist regulär; man kann endl. Automaten finden (keine weitere Erläuterung
  Zu 2: L ist nicht regulär, aber kontextfrei; Idee des Pumping-Lemmas für reguläre Mengen
  Zu 3: L ist deterministisch kontextfrei; Funktionsweise des Kellerautomaten erklärt
  Zu 4: L ist nicht-deterministisch kontextfrei
  Zu 5: L ist nicht regulär und nicht kontextfrei; Hinweis auf das Pumping-Lemma für kontextfreie Sprachen

- Was ist mit Pascal
  Ist wg. Variablendeklaration nicht kontextfrei (vgl. Sprache L_5)

- Was bedeutet $A \in ZEIT(n^2)$?
  Es gibt eine TM, die A in $O(n^2)$-Zeit erkennt.

- MUSS die TM immer halten?
  Ja!

- Ist A entscheidbar?
  Ja, wg. Definition!
- $A \in NZEIT(n^2)$
  Es gibt eine KTM M, die A in $O(n^2)$-Zeit erkennt.
- $L_M = \{x | \exists y \in \{0,1\}^*.\tau(x,y) = 1\}$
- Ist $A \in NZEIT(n^2)$ entscheidbar?
  Ja, aber nicht in n^2-Zeit, sondern $NZEIT(n^2) \subseteq \bigcup_c ZEIT(c^{n^2})$
- Was ist das P - NP - Problem?
  Gleichwertigkeit ist nicht geklärt, man weiß aber sicher, daß $P \subseteq NP$ gilt

Ich kann nur bestätigen, daß Prof. Weihrauch als Prüfer zu empfehlen ist. Die Prüfung verlief in sehr angenehmer Atmosphäre. Ich hatte immer genug Zeit zum Nachdenken und konnte bei den Beweisen auch mal zu einem zweiten Versuch ansetzen. Mit meiner Note (1,0) bin ich natürlich sehr zufrieden!

Vordiplomprüfung Theoretische Informatik

Kurs: **1653** und **1654**
Prüfer: Prof. Dr. Weihrauch
Datum: 31.05.1999
Dauer: ca. 25 Minuten

Theoretische Informatik A:

- Wann heißt eine Funktion berechenbar? (berechenbar durch eine k-stellige Registermaschine, p-rekursiv)
- Was bedeutet p-rekursiv? (Definition)
- Wie hängen berechenbare Wort- und Zahlenfunktionen zusammen? (Definition)
- Kann man aus der Berechenbarkeit einstelliger Zahlenfunktionen auch die Berechenbarkeit k-stelliger Funktionen ableiten? (Cantorische Tupelfunktion)
- Wie ist die Standardnumerierung φ definiert? (Definition)
- Was an der Standardnumerierung φ kann man als Programm auffassen?
- Welche zwei wichtigen Theoreme zur Standardnumerierung kennen Sie? (utm- und smn-Theorem mit Definition)
- Kennen Sie eine Numerierung der reellen Zahlen?
- Wie ist eine Numerierung definiert? (möglicherweise partielle, surjektive Fkt. $\nu:\subseteq N\rightarrow M$)
- Wie sieht es mit der Berechenbarkeit auf anderen Mengen (keine Zahlen, kein Alphabet) aus? (Numerierungen und induzierte Berechenbarkeit)
- Wann ist eine Menge $A\subseteq N$ rekursiv? (Definition)
- Wann ist eine Menge $A\subseteq N$ rekursiv aufzälbar? (alle drei Definitionen)
- Wie hängen diese Begriffe zusammen? (A rekursiv, gdw. A r.a. und MA r.a.)
- Ist A rekursiv eine echte Teilmenge von A rekursiv aufzählbar? (ja, Beweis durch $K\varphi$ r.a. aber nich rekursiv)
- Beweisidee warum $K\varphi$ nicht rekursiv ist. (genaue Erklärung warum aus $i\in N\backslash K\varphi <=> i\notin\varphi(i)$ folgt $N\backslash K\varphi$ nicht r.a.)
- Ist die Menge der Quadratzahlen rekursiv? (ja, Angabe Registermaschine)

Theoretische Informatik B:

- Zu welcher Sprachklasse gehören jeweils die folgenden Sprachen?
- $L1:=\{0^i1^j\}$ (regulär, Angabe eines endlichen Automaten)
- $L2:=\{w\$w^R|w\in\{0,1\}^*\}$ (kontextfrei, Angabe eines Kellerautomaten)
- $L3:=\{ww^R|w\in\{0,1\}^*\}$ (kontextfrei, Angabe einer Grammatik; erkennender Kellerautomat ist nichtdeterministisch)
- $L4:=\{w\$w|w\in\{0,1\}^*\}$ (weder regulär, noch kontextfrei, Beweis mit Pumping Lemma möglich)
- Ist PASCAL kontextfrei? (nein, Variabeln müssen deklariert werden)

Vordiplomsprüfung Theoretische Informatik (1653, 1645)

Prüfer: Hr. Prof. Dr. Weihrauch
Datum: 26.04.99
Note: 1,0

- Wann ist eine Zahlenfunktion berechenbar
- Ist x^2 berechenbar
- Zusammenhang zwischen eine Funktion „heißt“ und eine Funktion „ist“ berechenbar -> Churchsche These
- Datenmenge einer Bandmaschine mit Erklärung für (u,a,v)
- Befehlssatz einer Bandmaschine
- Kommt eine Bandmaschine ohne Hilfssymbole aus -> Hilfssymbollemma
- Zusammenhang zwischen Zahlen- und Wortberechenbarkeit
- Anforderungen an Modellsprache phi -> Anforderungen (U) und (S) erläutert, Hinweis auf utm- und smn-Theorem als formale Anforderung
- Definition von phi
- Berechenbarkeit auf Q
- Definition von (nrat, nrat) – berechenbar
- Ausweitung auf Berechenbarkeitsbegriff in R
- Definition Cauchy- Darstellung
- Typ-2-Maschine, Arbeitsweise erläutern
- Beispiele für Funktionen, die auf R berechenbar sind -> x^2, Wurzel(x), x+y
- Zwei Definitionen zu rekursiven Mengen
- Drei Definitionen zu rekursiv aufzählbaren Mengen
- Zusammenhang zwischen den Mengen
- Menge, die r.a. aber nicht rekursiv ist -> K phi
- Beweis, daß K phi r.a.
- Beweis, daß N\K phi nicht r.a.
- Ist die Menge der geraden Zahlen rekursiv
- Ist L={w aus {0,1}* mit 010 als Teilwort} regulär -> Angabe eines endlichen Automaten
- Ist L={wwR} kontextfrei -> Erkennung durch nichtdeterministen Kellerautomat

Komplexitätstheorie wurde nicht geprüft,

Vordiplom Theoretische Informatik
Kurs 1653/1654 Version 97/98

Termin: April 99
Prüfer: Prof. Dr. Weihrauch
Dauer: 25 min.
Note: 1,0

Wann ist eine Zahlenfunktion berechenbar?
Wenn es eine Registermaschine gibt, die diese Funktion berechnet. Auf die My-Rekursion brauchte ich nicht einzugehen.

Ist die Funktion $y = x^2$ berechenbar?
Ja, Registermaschine, welche x^2 berechnet.

Wann ist eine Wortfunktion berechenbar?
Wenn es eine Turing- oder Band-Maschine gibt, die diese Funktion berechnet.

Wie ist die Menge der Konfigurationen bei Bandmaschinen definiert?
(Q x D)

Wie ist die Datenmenge für Bandmaschinen definiert? Mit Erklärung der tatsächlichen Bedeutung dieser Definition.
D = (Gmma* x Gamma x Gamma*)
(u,a,v) Der Lesekopf steht über dem Zeichen a.

Gibt es einen Zusammenhang zwischen Wort- und Zahlenfunktionen?
Ja, über die Standardnumerierung Ny.
Eine Wortfunktion g ist berechenbar, gdw. Zugehörige Zahlenfunktion $f = ny^{-1} \circ g \circ ny$ berechenbar.

Wie ist die Standardnuemierung Phi definiert?

Nennen Sie die 2 wichtigen Sätze aus dem Skript.
Utm- und smn-Theorem

Was besagt die Forderung nach dem utm-Theorem?
Forderung (U) am Beginn Kap. 8

Wann ist eine Menge rekursiv bzw. rekursiv-aufzählbar?
Definition und Sätze dazu.

Zusammenhang zwischen rekursiven und r.a. Mengen?

Kennen Sie eine Menge die r.a. ist, aber nicht rekursiv? (mit Beweis)
K_{phi}

Wie ist die Berechnung in Q definiert?
Ny_{rat}-Darstellung von Q

Wann ist f: Q->Q (Ny_{rat},Ny_{rat})-berechenbar?
Definition
Was ist das Problem der Berechnung bei reellen Zahlen?

Reelle Zahlen nicht abzählbar.

Cauchy-Darstellung der reelen Zahlen

Wie ist die Berechenbarkeit von reellen Zahlen definiert?
(Rho,Rho)-Berechenbarkeit.

Typ-2-Maschine

Wie sind die Mengen P und NP definiert?

A Ì ZEIT(n^2). Ist A rekursiv?
Ja, da FZEIT nur totale Funktionen enthält?

B Ì NZEIT(n^2). Ist B rekursiv?
Ja da NZEIT(n^2) Ì Vereinigung von ZEIT(c hoch n^2) über alle c Element N

Sind die Sprachen A = {$a^n b^m$, n,m Î N} und B = {$a^n b^n$, n Î N} kontextfrei bzw. regulär?
A ist kontextfrei und regulär.
B ist nur kontextfrei, aber nicht regulär. Das kann man mit dem Pumpinglemma beweisen.

Regulärer Ausdruck für Menge A.
$a^* b^*$

Vordiplom Theoretische Informatik
Kurs 1653/1654 Version 98

Termin: 26.4.99
Prüfer: Prof. Dr. Weihrauch
Beisitzer: M. Schröder
Dauer: 30 min.
Note: 1,3

A Berechenbarkeit

Wann ist eine Zahlenfunktion berechenbar?

> Wenn es eine Registermaschine gibt, die diese Funktion berechnet. My Rekursion habe ich nicht erwähnt, wurde auch nicht nachgefragt.

Ist die Funktion y = x^2 berechenbar?

> Ja (zu genauerer Erklärung brauchte ich nicht mehr anzusetzen)

Überleitung zu Turing- und Bandmaschinen / Konfigurationen bei Bandmaschinen Wie ist die Datenmenge einer Bandmaschine definiert?

> D = (Gamma* x Gamma x Gamma*) und genaue Erläuterung der tatsächlichen Bedeutung dieser Definition: Tripel (u,a,v) mit u,a,v als Worte links, unter, rechts vom Schreib/Lesekopf.

Befehlssatz einer Bandmaschine

> L, R, t(a), f(a)

Gibt es einen Zusammenhang zwischen Wort- und Zahlenfunktionen? Zusammenhang zwischen ihrer Berechenbarkeit?

> Zusammenhang ueber Standardnumerierung Ny. Erklärung mit Diagramm: Wortfunktion g berechenbar, gdw. ihre zugehörige Zahlenfunktion ny- 1 g ny berechenbar.

Was ist klein-phi?

> Abbildung von N in P 1, Zusammensetzung der Funktion, Bedeutung (mehr generelle Erläuterung, zur genauen Definition kam ich nicht mehr)

UTM und SMN Theorem als Anforderung an Programmiersprache klein-phi: Was ist das Besondere an klein-phi?

> -hier verlor ich zum ersten Mal den Faden - wesentlich ist aber, daß es unendlich viele Numerierungen gibt, die das utm Theorem nicht erfüllen. Daraufhin habe ich von mir aus den Äquivalenzsatz erwaehnt/erklärt, was anscheinend o.k. war.

Warm ist eine Menge rekursiv bzw. rekursiv-aufzählbar? Welcher Zusammenhang besteht zwischen

Kennen Sie eine Menge die r.a. ist, aber nicht rekursiv?

Kphi - mit genauer Definition von Kphi und Beweis, wobei beim Beweis für r.a. die Erwähnung des utm Theorems schon genügte, der Beweis für die Nichtrekusivität (über N \ Kphi nicht r.a.) mußte dagegen genau sein.

Wie ist die Berechnung in Q definiert?

Definition und Erläuterung der Nyrat-Darstellung von Q

Was ist das Problem der Berechnung bei reellen Zahlen? Welcher MaschinenTyp wird benutzt?

Reelle Zahlen überabzählbar, d.h. es existiert keine Numerierung auf R, statt dessen wird mit Cauchy Dastellung ein x aus R durch unendliche Zahlenreihe "angenähert" ; Typ-2 Maschine erläutert

B Sprachen

Ist L={ w aus {0,1} *| 0 10 Teilwort von w } regulär?

Ja. Als regulärer Ausdruck {0 u 1}*. (010) . {0 u 1} * darstellbar.

Ist L={ wwR|w aus Sigma* } regulär?

Nein. Nicht regulär, aber nichtdeterministisch kontextfrei. Erklärung, daß kein Endlicher Automat konstruierbar ist, der wwR erzeugen kann, aber das Erkennen durch Kellerautomat möglich ist. Unterschied det./nicht det. Kellerautomat und L={ w$wR } als Beispiel für det. ktf. Sprache. Pumping Lemmata wurden nicht nachgefragt.

Regelmenge der kontextfreien Grammatiken in Chomsky Normalform?

C Komplexitaetstheorie

Welches ist der Zusammenhang zwischen Bandbedarf Sm(x) und Zeitbedarf Tm(x) -

Definitionen mit Beweisidee - beim Beweis der Abschätzung von Tm mußte ich passen

Wie ist NP Vollständigkeit erklärt?

Definition NP Vollständig mit Angabe Def. NP Hart, Pol-Reduzierbarkeit

Genereller Eindruck der Prüfung:

Insgesamt war die Atmosphäre während der Prüfung vollkommen ruhig und entspannt. Als Vorbereitung auf die Prüfung habe ich in der Hauptsache die Protokolle der letzten beiden Jahre gründlich durchgearbeitet.

W. wirft die Fragen nicht in den luftleeren Raum, sondern leitet sie immer mit einigen Worten ein. Trotzdem hatte ich bei 2 oder 3 Fragen Schwierigkeiten, sofort zu erkennen, worauf er abzielte, fand jedoch meistens nach entsprechenden Stichworten seinerseits recht schnell wieder den Faden. Ich hatte weder den Eindruck, hängen gelassen zu werden in einer solchen Situation, noch empfand ich, daß mir eben dieses "auf-dem-Schlauch-stehen" negativ angelastet wurde. Allerdings hatte ich schon das Gefühl, W. kann sehr genau sondieren, ob man sich denn tatsächlich mit dem Stoff

auseinandergesetzt hat oder nicht - auf stupides Auswendiglernen allein sollte man sich nicht verlassen, sondern schon eine gewisse Liebe zum Detail entwickeln (beispielsweise bei Definitionen von smn Theorem und rek. / r.a. Mengen verstehen, wann eine partiell rekursive und wann eine total rekursive Funktion notwendig ist).

Sehr geholfen hat mir, mit anderen zusammen Prüfung zu "spielen", also wirklich über das Thema zu reden. Erst dabei habe ich wirklich gelernt, genau und somit korrekt, die Fragen zu beantworten.