Gedächtnisprotokoll zur Prüfung Theoretische Informatik (Kurs 1657/1658)

Prüfer: Prof. Heinemann
Beisitzer: Dr. Rettinger
Note: 1,0
Datum: September 2011

Berechenbarkeit

- Wann heißt eine Zahlenfunktion berechenbar? (Registermaschine) Wann eine Wortfunktion? (Turingmaschine)
- Welche Operationen kann eine Registermaschine ausführen? (Inkrement, Dekrement, Test auf Null)
- Diese Operationen scheinen ja sehr begrenzte Möglichkeiten bereitzustellen. Wie kann man das Konzept erweitern? (verallgemeinerte Registermaschine)
- Es gibt noch andere Berechenbarkeitsmodelle, z.B. die μ-Rekursion. Wie ist diese definiert? (vgl. Kurstext, aufgeschrieben)
- Wann ist eine Menge entscheidbar, also rekursiv? (charakteristische Funktion berechenbar)
- Können Sie ein Beispiel nennen, ist z.B. die Menge aller Primzahlen rekursiv? Warum? (hier reichte eine Grundidee: alle Zahlen von 2 bis n als Teiler ausprobieren)
- Wann ist eine Menge rekursiv-aufzählbar? (Definitionsbereich einer berechenbaren Funktion)
- In welchem Zusammenhang stehen die beiden Begriffe? Warum? (jede rekursive Menge ist auch rekursiv-aufzählbar: bei Ergebnis 0 der charakteristischen Funktion in eine Endlosschleife laufen)
- Kennen Sie Mengen, die rekursiv-aufzählbar, aber nicht rekursiv sind? (Selbstanwendbarkeitsproblem, Halteproblem)
- Können Sie eines der beiden genannten Probleme definieren? (aufgeschrieben)
- Warum ist das Selbstanwendbarkeitsproblem rekursiv-aufzählbar? (hier habe ich etwas umständlich über das Halteproblem einen Zusammenhang zum Definitionsbereich von u_φ hergestellt – man konnte mir aber folgen, korrigierte mich wo nötig und ließ es dabei bewenden)
- Warum ist denn der Definitionsbereich von u_φ rekursiv-aufzählbar? (u_φ ist berechenbar, Querverweis zum utm-Theorem, das damit offenbar schon erledigt war)

- Warum ist das Selbstanwendbarkeitsproblem nicht rekursiv? (sein Komplement ist nicht rekursiv-aufzählbar)
- Können Sie beweisen, dass das Komplement des Selbstanwendbarkeitsproblems nicht rekursiv-aufzählbar ist? (Diagonalisierung über die Definitionsbereiche der berechenbaren Funktionen in $P^{(1)}$, diese unterscheiden sich alle an mindestens einer Stelle vom Komplement des Selbstanwendbarkeitsproblems – das Beweisverfahren musste ich nur andeuten, nicht im Detail erläutern; war außerdem wahrscheinlich keine Frage, die über Bestehen oder Nicht-Bestehen entschieden hätte...)
- Es gibt noch eine weitere wichtige Gruppe nicht rekursiver Mengen, kennen Sie die? (Satz von Rice)

Komplexität

- Welches wichtige Ergebnis über Komplexitätsklassen haben wir in diesem Teil des Kurses erhalten? (es können echte Teilmengenbeziehungen zwischen den Komplexitätsklassen gefunden werden – die Frage war irgendwie ein wenig anders formuliert, es war jedenfalls nicht wirklich schwer, auf die Antwort zu kommen)
- Wie können diese echten Teilmengenbeziehungen bewiesen werden? (Separations- und Hierarchiesätze)
- Können Sie einen der beiden Hierarchiesätze erläutern? (die vier Bedingungen für die Geltung des Zeithierarchiesatzes aufgeschrieben)
- Woher kommt die Bedingung $g \notin \mathcal{O}(f(n) \cdot log f(n))$ beim Zeithierarchiesatz? (daher, dass im Beweis eine Menge u.a. durch Diagonalisierung über die Sprachen in ZEIT(f) konstruiert wird, diese Diagonalisierung aber über Turingmaschinen mit zwei Arbeitsbändern geführt wird, die im Vergleich zu Turingmaschinen mit beliebig vielen Arbeitsbändern eine erhöhte Zeitkomplexität haben – ein tieferer Einstieg in diesen Beweis wurde nicht gefordert, aber die Grundidee sollte wohl verstanden sein)
- Können Sie das P-NP-Problem beschreiben? (echte Teilmengenbeziehung zwischen P und NP unklar)
- Was ist ein NP-vollständiges Problem? (vgl. Kurstext)
- Wenn man beweisen wollte, dass P = NP, wie könnte man die NP-vollständigen Probleme dafür benutzen? (finde einen deterministischen Polynomialzeit-Algorithmus für ein Problem M und zeige für ein beliebiges NP-vollständiges Problem, dass es sich auf M reduzieren lässt – dann lassen sich via Reduktion auf M alle Probleme in NP deterministisch in Polynomialzeit lösen)
- Was heißt NP-hart? (vgl. Kurstext)
- Wie funktioniert eine Reduktion zwischen zwei Mengen? (aufgeschrieben)

Formale Sprachen

- Wie sind die regulären Mengen definiert? (induktiv, genaue Definition vgl. Kurstext, aufgeschrieben)
- Von welchen Maschinentypen werden die regulären Sprachen erkannt? (endliche Automaten)
- Sind die endlichen Automaten deterministisch oder nichtdeterministisch? (gibt beide Typen)
- Erkennen die deterministischen und die nichtdeterministischen endlichen Automaten dieselben Sprachen? (ja) Warum? (weil sich jeder NEA über den Potenzautomaten in einen DEA überführen lässt)
- Von welchen Maschinentypen werden die kontextfreien Sprachen erkannt? (Kellerautomaten)
- Erkennen die deterministischen und die nichtdeterministischen Kellerautomaten dieselben Sprachen? (nein) Warum nicht? (die deterministisch-kontextfreien Sprachen sind unter Komplementbildung abgeschlossen, die nichtdeterministisch-kontextfreien nicht)
- Von welchen Maschinentypen werden die Typ-0-Sprachen erkannt? (nichtdeterministische Turingmaschinen)
- Erkennen die deterministischen und die nichtdeterministischen Turingmaschinen dieselben Sprachen? (ja) Warum? (weil sich jede NTM in eine DTM überführen lässt)

Fazit

Prof. Heinemann ist als Prüfer unbedingt zu empfehlen. Sehr ruhiger und freundlicher Stil. Keine unfairen Fragen. Definitionen und Zusammenhänge sollten klar sein, zudem legt er offenbar Wert auf eine präzise Ausdrucksweise (es ist z.B. ein Unterschied, ob die charakteristische Funktion existiert (tut sie immer), oder ob sie berechenbar ist!). Aufschreiben musste ich aber relativ wenig. Ich wurde nur zweimal explizit nach Beweisideen gefragt, hatte aber den Eindruck, dass er auch so merkt, ob man die zentralen Beweise im Kurstext nachvollzogen und verstanden hat.

Prüfungsprotokoll

Mündliche Prüfung (Bachelor Informatik)
Einführung in die Theoretische Informatik (1657+1658)
Datum: Montag, 26.9.2011
Prüfer: Prof. Heinemann

Ich habe die Prüfung gerade so bestanden und kann bestätigen, dass es nicht genug ist, Definitionen auswendig zu lernen. Herr Heinemann will auf jeden Fall klare Definitionen hören, aber er stellt dann auch weitergehende Fragen, um zu sehen, ob man die Materie wirklich verstanden hat. Hier ein kurzer Überblick über seine Fragen:

1. Was für Operationen kann eine Registermaschine in einem Schritt ausführen?
2. Welche Zahlenfunktionen sind berechenbar?
3. Wie ist der Zusammenhang zwischen Wort- und Zahlenfunktionen bezüglich der Berechenbarkeit?
4. Wie sind rekursive Mengen definiert? - Hier machte ich den Fehler, dass ich sagte, dass die charakteristische Funktion **definiert** sein muss. Sie muss aber natürlich **berechenbar** sein!
5. Wie sind rekursiv-aufzählbare Mengen definiert?
6. Wie verhalten sich rek. Mengen und r.a. Mengen zueinander?
7. Welche Menge ist nur r.a., aber nicht rekursiv? → Das Halteproblem/Selbstanwendbarkeitsproblem Kphi
8. Bitte notieren Sie Kphi.
9. Wie zeigen wir, dass Kphi r.a. Ist? → mittels des utm-Theorems (diesen Zusammenhang konnte ich z.B. nicht korrekt erklären)
10. Wie lautet der Zeithierarchiesatz?
11. Was ist NP?
12. Was bedeutet NP-vollständig?
13. Was bedeutet NP-hart?
14. Was bedeutet es denn genau, dass eine Menge auf eine andere Menge polynomiell reduzierbar ist? → Da hatte ich nur eine ungefähre Vorstellung, aber die Antwort muss man exakt formulieren können.
15. Was sind die regulären Mengen?
16. Wie entscheidet man, ob eine Sprache kontextfrei ist? → mittels Pumping -Lemma
17. Skizzieren Sie das Pumping-Lemma.
18. Kann man dieses Pumping-Lemma auf alle Sprachen anwenden? → Die richtige Antwort wäre nein gewesen, denn wie ich dann erfuhr, ist die Erfüllung nur eine hinreichende, aber nicht notwendige Bedingung dafür, dass die Sprache L kontextfrei ist. Man kann somit aus dem Lemma nur schließen, dass eine Sprache NICHT kontextfrei ist, aber nicht beweisen, dass sie kontextfrei ist.

Wie man anhand der Übersicht sehen kann, bin ich bei einigen Fragen ziemlich ins „Schwimmen“ geraten. Ich hatte zur Prüfungsvorbereitung ein paar Mal mit einem Kommilitonen geskypt, und diese Form des Lernens kann ich nur empfehlen. Das hätte ich mal intensiver nutzen sollen! Herr Heinemann meinte noch, dass man auch unbedingt die Übungsklausuren und auch die Betreuungsangebote in den Studienzentren nutzen sollte.
Mein Fazit: Herr Heinemann ist ein netter Prüfer, aber ich habe mich mit der Materie einfach sehr schwer getan, und dass hat er dann auch gemerkt. Die Prüfung erfordert auf jeden Fall eine äußerst intensive Vorbereitung. Aus meiner Sicht der bisher mit Abstand schwerste Kurs in meinem Studium.

Gedächtnisprotokoll Theoretische Informatik 1657/1658

Datum:4.2.2010
Prüfer: Herr Prof. Heinemann
Beisitzer: Dr. Rettinger

Professor Heinemann ist kein Mensch der lange Einführungsreden hält. Wir fingen nach 2-3 Sätzen gleich an.

1. Wann heißt eine Funktion berechenbar?
2. Wir haben die verallgemeinerten Registermaschinen kennengelernt. Was ist das? (Definition)
3. Gibt es noch andere Berechenbarkeitsmodelle?
4. Ist die µ- Rekursion in die Registerberechenbarkeit gleichwertig?
5. Wie ist die µ-Rekursion definiert?
6. Mit welchem Maschinentyp werden die Wortfunktionen berechnet?
7. Es gibt dort einen Zusammenhang zwischen Zahlenfunktionen und Wortfunktionen. Welcher ist das? (Standardnummerierung über Sigma)
8. Wir haben eine Nummerierung über P kennengelernt. Was ist P?
9. Warum können wir uns auf P hoch 1 beschränken? (Cantor)
10. Wir haben zwei wichtige Eigenschaften einer Programmiersprache behandelt. U und S. Was ist U?
11. Ist Uphi partiell? Und warum?
12. Nennen sie ein Beispiel einer partiellen Funktion? (Die überall nirgends definierte Funktion).
13. Dann hatten wir noch einen wichtigen Satz über gross Phi (im Zusammenhang mit UTM und SMN Theorem) kennengelernt. Wie lautet dieser? (Gross Phi-Theorem)
14. Jetzt zu Mengen: Wie lautet die Definition eine Menge ist rekursiv aufzählbar?
15. Der Def(Uphi) ist eine bestimmte Menge welche?
16. Was ist das besondere an dieser Menge?
17. Warum ist sie rekursiv aufzählbar? (Gerade weil sie dem Def(Uphi) entspricht.
18. Nennen sie den Satz von Rice!
19. Kommen wir jetzt zur Komplexitätstheorie. Was ist O(g)?
20. Was ist Zeit(f)?
21. Kann man eine echte Teilmenge innerhalb der Zeitkomplexitätsklassen bilden?
22. Antwort Ja. Wie funktioniert das? (Zeithierachie- und Seperationssätze)
23. Wie ist der Zeithierachiesatz definiert.
24. Kommen wir nun zu den Sprachen. Hier hatten wir die Chomskyhierachie kennengelernt. Was können sie mir dazu sagen. (Typ0, Typ1,…)
25. Was sind die Typ0-Sprachen?
26. Wie lautet die Regelmenge der Typ1-Sprachen?
27. Wie lautet die Regelmenge der Typ 2-Sprachen?
28. Wie lautet die Regelmenge der regulären Sprache?
29. Welchen Maschinentyp haben wir bei den Typ- 3 Sprachen kennengelernt?
30. Welche Sprache wird bei diesem Maschinentyp im deterministischen und nichtdeterministischen Fall erkannt? Sind diese Maschinen gleichwertig?
31. Welchen Maschinentyp haben wir bei den Typ-2 Sprachen kennengelernt?
32. Wie verhält sich hier der deterministische und nichtdeterministische Maschinentyp?

Ich kann Prof. Heinemann empfehlen. Er fragt direkt und erwartet kurze präzise Antworten.
Lange Ausführung sollte man vermeiden. An seinem Gesichtsausdruck habe ich vernommen, dass er
Damit nicht so zufrieden ist. Definitionen sollten sehr genau wiedergegeben werden.
Wenn man eine Frage nicht beantworten konnte, wird auch nicht lange darauf rumgeritten. Es beginnt dann sofort mit der nächten.
Ich wünsche euch Viel Glück.

Prüfungsprotokoll

Bachelorprüfung Theoretische Informatik (25310)
Kurse: 01657 – Grundlagen der theoretischen Informatik A
01658 – Grundlagen der theoretischen Informatik B

Datum:	27.08.2009
Prüfer:	Prof. Dr. Heinemann
Beisitzer:	Dipl.-Inf. Ruth Dillhage
Dauer:	30 Minuten
Note:	1,7

Theoretische Informatik A

- Wann sind Wortfunktionen berechenbar?
- Was ist eine Turingmaschine?
- Was ist eine Bandmaschine?
- Was besagt das Hilfssymbollemma?
- Wofür benötigt man die BM? → φ
- Was ist φ? Für was ist es eine Nummerierung?
- Was sind das utm- und snm-Theorem?
  → utm aufschreiben
- Wann ist eine Funktion rekursiv?
- Welche Funktionen sind nicht rekursiv? Beispiel?
  → Äquivalenzproblem
- Was ist das Halteproblem?
  → Definitionsmenge von φ
- Ist das Halteproblem rekursiv?
  → nein, nur rekursiv-aufzählbar
- Was ist die charakteristische Funktion?
- Was ist die Typ-2-Maschine
  → Haben wir übersprungen, da ich keine Ahnung hatte

Theoretische Informatik B

- Was bedeutet ZEIT(f)? Was ist dabei f?
- Was ist NP?
- Welches NP-vollständige Problem kennen Sie?
  → SAT
- Was ist SAT?
- Was bedeutet NP-vollständig?
- Was bedeutet NP-hart?
- Wo wird die NTM benötigt? → Typ0-Sprachen
- Welche TM wird für Typ1-Sprachen benutzt?
- Sind NTM und DTM gleichmächtig?
- Warum benötigt man eine eingeschränkte TM für die Berechenbarkeit und was ist an ihr anders als in der TM aus Kurs 1?

Fazit:
Prof. Dr. Heinemann ist unbedingt als Prüfer zu empfehlen. Er geht die Prüfung entspannt an und hilft einem mit Hinweisen bzw. formuliert die Fragen anders wenn man hängt.
Er möchte sehen, dass man das Thema verstanden hat, man muss aber wichtige Definitionen wiedergeben können.
Die älteren Prüfungsprotokolle (auch für die Diplomprüfungen) waren sehr hilfreich für mich, es hilft einem aber nichts, diese auswendig zulernen und zu hoffen, dass das reicht!
Ich war positiv überrascht über die gute Note, da ich öfter Hänger hatte und mich besonders im zweiten Kurs nicht sicher gefühlt habe.
Auch kamen ein paar Fragen dran mit denen ich nicht gerechnet hatte und mit denen ich nichts anfangen konnte.

Prüfung: Grundlagen Theoretische Informatik A und B (Bachelorprüfung)
Prüfer: Prof. Heinemann
Termin: 19.12.08
Note: 1,7
Dauer: 25 min

Fragen Berechenbarkeit (Teil A):
-Standardnummerierungen: wozu braucht man sie, welche kennen Sie?
-Wann ist eine Menge v-rekursiv
-Beziehung Wortfunktionberechenbar, Zahlenfunktionberechenbar
-UTM-Theorem (U_phi-Definition)
-Ist der Definitionsbereich von U_phi entscheidbar?
-> Hier kam ich einfach nicht darauf, dass damit
die Menge K_phi_0 gemeint war, diese ist rekursiv aufzählbar und natürlich nicht rekursiv=entscheidbar,
ich stand dann auf dem Schlauch und antwortete falsch entscheidbar...
-Welche Mengen kennen sie, die nicht rekursiv aufzählbar ist?
-> Das Komplement von K_phi
-Kennen sie das Äquivalenzproblem oder das Korrektheitsproblem?
->hier musste ich passen, steht alles im Kurstext drin
-Nennen sie den Satz von Rice

Fragen Komplexitätstheorie (Teil B):
-Definieren sie ZEIT
-Was heißt NP-vollständig

Prof. Heinemann ist ein sehr netter Prüfer, der die Prüfung langsam anfängt mit einleitenden Worten zu Kursen und bei Unsicherheiten die Fragen gerne anders und behutsam formuliert.
Ich kann ihn nur wärmstens empfehlen. Bei mir wurden keine Beweise gefragt, Definitionen und Sätze
mussten genau angegeben oder hingeschrieben werden. Die Zusammenhänge zwischen den Definitionen
sollten gut verstanden sein. Ansonsten kann man mit den Prüfungsprotokollen aus den Vordiplomsprüfung
1653/54 sehr gut lernen.