

Bachelor Informatik: Modulprüfung **1810 Übersetzerbau**
Gedächtnisprotokoll aus 2012

Prüfer: Prof. Dr. Güting

Dauer: 25 min

Note: 1.0

Allgemein

Generell ist Prof. Dr. Güting ein ruhiger und sehr freundlicher Prüfer, der auch über kleine Unsicherheiten hinwegsieht. Wenn man nicht sofort auf eine Frage eine Antwort weiß, versucht Prof. Dr. Güting einen mit Tipps in die richtige Richtung zu lenken. Alles in Allem war ich mit der Prüfung sehr zufrieden. Da ich wirklich ab und an Probleme hatte und mir Prof. Dr. Güting auf die Sprünge helfen musste, war ich über die Note positiv überrascht. Die Prüfung fing ca. eine halbe Stunde später an als vereinbart, das hat soweit aber nicht gestört.

Wie aus anderen Prüfungsprotokollen (nicht nur 1810) ersichtlich wird, legt Prof. Dr. Güting relativ viel Wert auf Definitionen und korrekte mathematische Notationen. Zumindest kommt mir das so vor. Ich hatte bis jetzt aber auch erst eine mündliche Prüfung bei einem anderen Prüfer. Man sollte sich deshalb auf jeden Fall die wichtigsten Definitionen (DEA, kontextfreie Grammatik, start, FIRST, FOLLOW, Steuermengen, ...) in korrekter mathematischer Notation merken. Bei mir wurde bspw. bei der Definition der Produktionen der kontextfreien Grammatik bemängelt, dass ich $N \rightarrow (N \cup E)^*$ anstatt $N \times (N \cup E)^*$ geschrieben habe. Da ich nach entsprechenden Hinweisen von Prof. Dr. Güting auf die korrekte Notation gekommen bin, hat sich das aber scheinbar nicht auf meine Note ausgewirkt.

Fragen

Was macht ein Übersetzer?

Für was werden Übersetzer eingesetzt?

Was ist eine kontextfreie Grammatik (Definition)?

Kann man mit einem Übersetzer bspw. auch Englisch nach Japanisch übersetzen?

Welche Übersetzungsphasen gibt es?

Was ist die Postfix-Notation?

Wie wird die Postfix-Notation ausgewertet?

Was ist 3AC?

Welche 3AC-Befehle kennen Sie?

Wie könnte eine while-Schleife in 3AC übersetzt werden?

Welche Verfahren gibt es bei der Syntaxanalyse?

Wie wird die Bottom-up-Analyse noch genannt?

Was passiert beim shift, was beim reduce?

Wie funktioniert die Top-down-Analyse?

Wie wird das Verlaufen in Sackgassen verhindert?

Wie lautet die Definition von FIRST?

Was ist die FOLLOW-Menge?

Warum wird die FOLLOW-Menge überhaupt benötigt?

Wie kann die Top-down-Analyse implementiert werden?

Können Sie einen vorgreifenden Analysator als abstrakte Maschine zeichnen?

Wie funktioniert die Top-down-Analyse mit Analysetabelle genau, was passiert in der Maschine?

Gedächtnisprotokoll zur Prüfung Übersetzerbau (Kurs 1810)

Prüfer: Prof. Güting

Note: 1,0

Datum: April 2011

Die Prüfungsinhalte waren ähnlich den anderen auf der Fachschaftsseite protokollierten Prüfungen:

- Warum übersetzt man? Einfacher in Sprache A zu schreiben, benötigt wird aber Sprache B.
- Welche Anwendungsgebiete gibt es? Programmier-, Dokumentbeschreibungs-, Datenbankabfragesprachen, VLSI etc.
- Phasen der Übersetzung? Analyse und Synthese; lexikalische Analyse, Syntaxanalyse, semantische Analyse, Zwischencodeerzeugung, Codeoptimierung, Codeerzeugung (die drei Phasen der Analyse jeweils mit Ein- und Ausgabe und kurzer Erläuterung der Arbeitsschritte). Von hier aus Vertiefung der lexikalischen und der Syntaxanalyse.
- Reguläre Sprachen (Definition).
- Darstellung regulärer Ausdrücke als Zustandsdiagramme (Stellen Sie einen typischen Bezeichner einer Programmiersprache als regulären Ausdruck und als Zustandsdiagramm dar! Bsp. `letter(int|letter)*`).
- Durchführung der lexikalischen Analyse (Durchgehen von Zustandsdiagrammen usw.).
- Kontextfreie Grammatiken (Definition).
- Verschiedene Möglichkeiten für die Syntaxanalyse (Top down und Bottom up), jeweils kurze Erläuterung.
- Vertiefung der Top down-Analyse, genauere Erklärung (paralleles Weitergehen durch Eingabe und bisher erzeugten Syntaxbaum, Abgleich der aktuellen Symbole und entsprechende Aktionen).
- Linksfaktorisierung und Beseitigung von Linksrekursion.
- Prinzip des predictive parsing, kurze Erwähnung von LL(k)-Grammatiken (ohne weitere Erläuterung).
- Erklärung der Mengen start, FIRST und FOLLOW und der Steuermengen (Definitionen).
- Was macht man dann mit diesen Mengen? Erstellung einer Analysetabelle, die für die Auswahl der richtigen Produktion genutzt wird.

- Erläuterung des Parsing-Vorgangs, vgl. abstrakte Maschine im Kurstext (Eingabe, Ausgabe, Analysetabelle, Stack).
- Nicht eingegangen wurde auf Zwischensprachen, Codeerzeugung, Codeoptimierung, Speicherorganisation, attributierte Grammatiken und Übersetzungsschemata, konkrete Übersetzung von Dokumentbeschreibungs-, imperativen und funktionalen Programmiersprachen. Also nichts aus den Kurseinheiten 4 bis 7 – daraus allgemeine Rückschlüsse zu ziehen, ist aber wohl nicht unbedingt zu empfehlen.
- Ich habe für die Prüfung die relevanten Definitionen (kontextfreie Grammatik, FIRST und FOLLOW usw.) auswendig gelernt und diese an passenden Stellen auch ohne explizite Aufforderung aufgeschrieben oder dies zumindest angeboten. Das ist meinem Eindruck nach ganz gut angekommen. Außerdem sollte man die Beseitigung von Linksrekursion und die Linksfaktorisierung beherrschen (Beispiele konnte ich selbst aussuchen). Die diversen Parsing-Techniken sollten wohl mindestens (!) der Idee nach verstanden und präsent sein. Daneben ist natürlich insgesamt ein guter Überblick über die Kursinhalte und die Zusammenhänge hilfreich ;-)

Prüfungsprotokoll der mündlichen Prüfung Übersetzerbau 1810 (Bachelor Informatik)

Prüfer: Prof. Dr. Güting

Semester der Prüfung: WS 09/10

Datum der Prüfung: 07.06.2010

Dauer: ca. 35 min

Note: 2.0

Hier sind sicherlich nicht alle Fragen, jedoch ein großer Teil.

Allgemein:

Warum übersetzt man überhaupt?

Was für Sprachen kann man übersetzen ?

Was ist eine kontextfreie Grammatik ?

-> Erklärung und Definition

Was sind die logischen Schritte einer Übersetzung ?

-> Alle Phasen genannt und kurz erklärt

Was ist der Unterschied zwischen Compiler und Interpreter ?

-> Erklärt und die Vor- und Nachteile genannt

Syntaxanalyse:

Welche Verfahren gibt es ?

-> Top-Down Analyse und Bottom-Up Analyse kurz erläutert

Top-Down Analyse:

Welche Probleme gibt es ?

-> Sackgasse, Linksrekursiv...

Was passiert genau auf dem Stack ?

Was ist hier eine Abstrakte Maschine ?

Wie wird eine Analysetabelle erstellt ?

Was macht ein Vorgreifender Analysator ?

Wie wird die Steuermenge bestimmt

-> Erklärt und FIRST und start k Definitionen aufgeschrieben

Bottom-Up Analyse:

Was ist ein Shift-Reduce-Parser ?

Welche Probleme gibt es ?

-> Handle erkennen und Definition aufgeschrieben

Welches ist das einfachste Verfahren ?

-> Operator Vorrang-Analyse erklärt mit Tabelle und Stack

Was passiert genau auf dem Stack ?

-> Erklärt...

Zur Prüfung allg.:

Prof. Dr. Güting erwartet ,das die wichtigen Definitionen sitzen. Insgesamt war die Prüfung recht angenehm.

Mündl. Bachelor-Prüfung 1810 "Übersetzerbau"
Februar 2006
Prüfer: Prof. Güting
Note: 1.3

* Warum übersetzt man überhaupt?

- > Überführung von einer Sprache A nach B. Kann u.U, sinnvoller sein,
- > etwas in A auszudrücken, als in B.

* Prinzipieller Ablauf eines Übersetzers

- > 6 Phasen, einzelne Phasen kurz erläutert
- > Rückfrage: Was passiert global über allen Phasen: Fehlererkennung und
- > noch was zweites, das ich wieder vergessen habe. Konnte die Frage nicht wirklich beantworten.

* Unterschied Compiler - Interpreter

* Lexikalische Analyse

- > Grundlagen: reguläre Ausdrücke, formale Sprachen. Sollte die formale Definition erläutern können.
- > Was ist eine Sprache? (nicht unbedingt eine formale Sprache, ganz allgemein). Hier hatte ich ein paar Schwierigkeiten, aber nach ein paar Hilfestellungen kam ich drauf.
- > DEA erklären: erkennt reguläre Ausdrücke Implementierung DEA per
- > Hand:
Zustandsdiagramme implementieren (nur kurs angedeutet)
- > Implementierung DEA per Lex:
Lex überführt die Spezifikation in C-Quellcode. Die Lex-Spezifikation enthält C-Code (z.B. return TOKEN). Token werden an den Parser weitergereicht.
Mein Beispiel (nicht wirklich Lex): RegExp: NUMBER [0-9]* -> return NUMBER
- > Rückfrage: Und wie kommt die Zahl selbst in den Scanner: ylval

* Syntaxanalyse

- > beide Verfahren kurz erläutert
- > konkreter: Bottom-Up-Analyse: allg. Ablauf, Stack, 4 Befehle
- > Frage: Was ist das einfachste Verfahren? Operator-Vorrang-Analyse, erläutert: Operatoren, Operator-Tabelle, Erkennen von Handles

* Attributierte Grammatiken

- > Formal erklären (Definition Attributierte Grammatiken, Semantische Regeln)
- > synthetisierte & vererbte Attribute erklären und unterscheiden
- > S-attributierte Definition & L-attributierte Definition erklären & unterscheiden

* Fazit:

Der Kurs ist kompakt und daher sehr prüfungsfreundlich. Das wurde auch schon in einem anderen Protokoll erwähnt.
Prof. Güting ist kein besonders guter "Small-Talker". Andere Professoren können durch einen kleinen Plausch vor der Prüfung einem die Prüfungsangst wenigstens ein bisschen nehmen. Tipp: Fangt selber mit dem Small-Talk an. Meiner Erfahrung nach steigt Hr. Güting dann gerne mit ein. Im Allgemeinen ist er ein korrekter & fairer Prüfer. Daher kann ich ihn als Prüfer auch nur empfehlen.

Kurs: 1810
Prüfer: Prof. Güting
Datum: 25.05.05
Dauer: 25min
Note: 1,7

Die Prüfung verlief offensichtlich ähnlich wie die hier ebenfalls protokollierte Prüfung vom 07.01.2004. Trotzdem wiederhole ich die Fragen hier:

- Was tut ein Übersetzer?
- Was übersetzt er?
- Welche Sprachen eignen sich als Quellsprachen? Hier wollte er die exakte Definition von kontextfreien Grammatiken haben
- Welche Phasen unterscheidet man?
- Welche Zwischensprachen kennen Sie?
- Wie wird Postfix-Notation verarbeitet?

Operanden werden auf den Stack gepackt, sobald ein Operator in der Eingabe erscheint, werden die Operanden vom Stack genommen, die Operation ausgeführt und das Ergebnis wieder auf den Stack gelegt.

- Welche 3AC-Befehle kennen Sie? Hier sollte man wirklich alle Gruppen kennen.
- Welche Möglichkeiten hat man, die Top-Down-Analyse mit Vorausschau durchzuführen? Mit Analysetabelle und mit rekursivem Abstieg.
- Wie erstellt man die Analysetabelle?

Hier wollte er auch wissen, aus was für einem Automaten man die Analysetabelle gewinnt. Bin dabei auf die Steuermengen zu sprechen gekommen. Habe die genaue Definition aufgeschrieben.

- Was sind denn $FIRST_k(\alpha)$ und $start_k$? wollte er in dem Zusammenhang noch wissen

$FIRST_k(\alpha)$ habe ich noch korrekt definiert, $start_k$ nur noch informell.

- Und wozu brauche ich bei der Definition der Steuermengen noch die *FOLLOW*-Mengen?

Weil es sein kann, dass man aus $\alpha \varepsilon$ ableiten kann

- Prof. Güting zog eine vorbereitete Karteikarte hervor, auf der in etwa folgendes stand:

$$i_1 : A \rightarrow \alpha_1 | D$$

$$i_2 : A \rightarrow \alpha_2 | D$$

Hieraus sollte ich eine Procedure in Pseudocode schreiben, die den rekursiven Abstieg implementiert.

- Welche Arten von Grundsymbolen hat eine Programmiersprache üblicherweise?
- Welche Arten der Syntaxanalyse gibt es?

Das war's so im Großen und Ganzen.

Bemerkungen: Über Kurseinheit 2 sind wir damit kaum hinausgekommen. Die Bewertung fand ich sehr fair, da ich an einigen Stellen gewisse Schwierigkeiten (z. B. bei der Implementierung des rekursiven Abstiegs). Die Fragen sind recht offen formuliert, so dass man auch mal einfach erzählen kann. Hier unterbricht er einen auch nicht.

Übersetzerbau (01810)

Prüfer: Herr Prof. Güting
Dauer: ca. 25 Minuten, 27.4.2005

Note: 2,0

- Was macht ein Übersetzer ?

Übersetzt von Sprache A nach Sprache B. Grund: Mittels Sprache A ist Problemstellung besser beschreibbar, jedoch existiert nur Maschine/Anwendung, die Sprache B versteht.

- Kann man auch natürliche Sprachen wie Deutsch oder Japanisch übersetzen ?

Nein, da diese Sprachen keine kontextfreie Grammatik haben.

- Was ist der Unterschied zwischen Compiler und Interpreter ?

Compiler: Übersetzt gesamte Eingabe, Ausführung des Programms erst zu einem späteren Zeitpunkt.

Interpreter: Übersetzt immer nur einen kleinen Teil der Eingabe (z.B. eine Zeile) und führt sofort Befehl aus.

Vorteil Compiler: Schnellere Ausführungszeit; Vorteil Interpreter: schnellere Programmentwicklung

Mischform: Compiler übersetzt Programm in einfachen Zwischencode, der zur Laufzeit effizient von einem Interpreter verarbeitet werden kann (Beispiel Java)

- Welche Phasen durchläuft ein Übersetzer ?

6 Phasen (jeweils die Besonderheiten mit Ein-/Ausgabe erläutert): lexikalische Analyse, syntaktische Analyse, semantische Analyse, Zwischencode-Erzeugung, Optimierung des Zwischencodes, Erzeugung des Maschinencodes.

- Was ist 3-Adress-Code ?

einfache Zwischensprache mit maximal drei Operatoren, z.B. binäre Operationen, Zuweisungen, Arrayzugriffe, bedingte und unbedingte Sprünge. 3AC ist abstrakt genug, um für eine Optimierung geeignet zu sein, aber andererseits so einfach, dass 3AC-Befehle durch einen oder wenige Maschinen-Code-Befehle ersetzt werden können.

- Welche Arten der Syntaxanalyse gibt es ?

Top-Down: Erstellung des Ableitungsbaum von der Wurzel her zu den Blättern durch Auswahl der richtigen Ableitung

Bottom-Up: Die Blätter werden von "unten" zu größeren Strukturen bis zur Wurzel zusammengesetzt, Erkennung von "Handles"

- Welche Probleme können bei der Top-Down-Analyse entstehen ?

Problem ist die Auswahl der richtigen Ableitung. Durch Verlaufen in "Sackgassen" wird die Übersetzung ineffizient, durch Linksrekursion wird die Analyse unmöglich.

- Was macht man bei Linksrekursion ?

Linksrekursion kann immer aus der Grammatik entfernt werden. Ich sollte nun an einem einfachen Beispiel demonstrieren, wie man konkret die Linksrekursion entfernt. Das habe ich aber irgendwie nicht hinbekommen und mich ziemlich verheddert, was sich leider negativ auf die Note bemerkbar machte.

- Was macht man, um das Verlaufen in "Sackgassen" zu vermeiden ?

Linksfaktorisierung (an einem Beispiel erläutert), dann Erstellung von Steuermengen für Produktionen, so dass eine vorausschauende Auswahl der richtigen Ableitung möglich ist.

- Können Sie für diese Grammatik hier (zeigt eine Karteikarte mit einfacher Grammatik) die Steuermengen erstellen ?

Erstelle Steuermengen für die Produktionen.

- Wie kann man das formal darstellen ?

Gebe Definitionen für FIRST- und FOLLOW-Mengen.

Allgemeiner Eindruck:

Prof. Güting war sehr freundlich. Er meinte, dass mir die allgemeine Darstellung der Zusammenhänge sehr gut gelungen sei, wenn es jedoch technischer wurde (Eliminierung der Linksrekursion, formale Definition der Steuermengen etc.) ich jedoch Unsicherheiten gezeigt hätte. Ich selbst empfinde die Benotung als gerecht.

Wie bereits in den anderen Prüfungsprotokollen angedeutet wurde, ist KE 6 (Funktionale Sprachen) nicht prüfungsrelevant, das hatte ich mir bei der Prüfungsanmeldung noch einmal bestätigen lassen.

Gedächtnisprotokoll „Übersetzerbau“ (Kurs 1810)

Prüfer: Prof. Dr. Güting
Datum: 7.1.2004

Die Prüfung war insgesamt angenehm und ziemlich kurz.
Prof. Güting ist ein netter und freundlicher Prüfer.

Zu Anfang stellte Prof. Güting eine ganze Reihe allgemeiner, einleitender Fragen, bei denen z.T. ich Schwierigkeiten hatte, zu verstehen, was genau er von mir hören wollte:

- Was tut ein Übersetzer?
- Was übersetzt er?
- Welche Sprachen eignen sich als Quellsprachen? (Stichwort: kontextfreie Grammatiken)
- Welche Phasen unterscheidet man?
- Was ist eine Zwischensprache?
- Welche Zwischensprachen kennen Sie?
- Wie sehen 3AC-Befehle aus?
- Welche 3AC-Befehle kennen Sie?

Dann kamen wir auf die lexikalische Analyse zu sprechen:

- Welche Arten von Grundsymbolen hat eine Programmiersprache üblicherweise?
- Wie lassen sich die Grundsymbole einer Quellsprache beschreiben?
- Was sind reguläre Ausdrücke, was reguläre Sprachen? (genaue Definition gefordert)
- Wie lassen sich die Grundsymbole erkennen? (endliche Automaten)
- Was ist ein endlicher Automat?
- Wie funktioniert er?
- Wie erkennt man damit die Grundsymbole? Wie analysiert man die gegebene Eingabezeichenfolge lexikalisch? (genaue Beschreibung des Algorithmus)

Als nächstes war die Syntaxanalyse dran:

- Welche Arten der Syntaxanalyse gibt es?
- Beschreiben Sie die Top-Down-Analyse möglichst genau!
- Was ist die Besonderheit des predicitive parsing?

Und das war's dann auch schon.

Im Nachgespräch sagte Prof. Güting, dass er das Kapitel zu funktionalen Programmiersprachen und deren Interpretation und Übersetzung bislang nie geprüft habe und künftig den Kursteilnehmern auch mitteilen möchte, dass die Bearbeitung dieses Kapitel freiwillig sei, aber empfohlen werde.

Gedächtnisprotokoll zur mündlichen Diplomprüfung

Kurs: 1810 Übersetzerbau

Datum: 11.4.2001

Prüfer: Prof. Güting

Beisitzer: Stephan Dieker

Dauer: ca. 25 min

Note: 1.0

Fragen:

- Wofür werden Compiler eingesetzt? Ich habe dann die Beispiele aus dem Kurs aufgezählt, klassischerweise eben Übersetzung einer höheren Programmiersprache in Assembler / Maschinensprache, ansonsten Dokumentenbeschreibung etc. . Anmerkung von Prof. Güting: "Eben alles, wo man syntaktische Strukturen erkennen kann".
- In welche logischen Phasen wird ein Compiler gegliedert? Ich hab dann gesagt, daß es eine Grobeinteilung in Analyse und Synthese gibt und dann die 6 Phasen mit Ein- und Ausgabe aufgezählt.
- Wie werden die Token in der lexikalischen Analyse beschrieben? -> reguläre Ausdrücke
- Was wird mit den Token beschrieben? Schlüsselwörter, Operatoren, Bezeichner, Konstanten
- Was sind reguläre Ausdrücke? Hier mußte ich erst mal die Definition aus dem 1654 wieder aus dem Gedächtnis kramen und habe mich so durchgeholpert, was ihm aber reichte. Er meinte noch, man müsse da eigentlich zwischen regulären Ausdrücken und regulären Mengen sauber unterscheiden.
- Wie werden sie erkannt? endlicher Automat
- Wie sieht so ein endlicher Automat aus?
- Wie implementieren sie das von Hand?
- Was machen sie, um es nicht von Hand implementieren zu müssen? Ich verwende einen Scannergenerator wie z. B. lex.
- Wie funktioniert das? Wirkungsweise von lex erklärt: Erzeugt aus regulären Ausdrücken und C-Programmstücken ein C-Programm, daß den DFA implementiert. Ihm war noch wichtig, daß in der Aktion der Wert des Token mit return an den Parser zurückgegeben wird.
- Welche Parse-Strategien kennen sie? Bottom-Up und Top-Down
- Erläutern sie die beiden mal kurz! Hier habe ich wirklich kurz erläutert, daß ich im ersten Fall die Eingabe auf einen Stack schiebe und versuche, Handles zu erkennen und bis zum Startsymbol zu reduzieren, im anderen Fall vom Startsymbol aus losgehe und versuche, die Eingabe zu erzeugen.
- Welches ist der einfachste Bottom-Up-Parser, den sie kennen? Operator-Präzedenz-Parser
- Erläutern sie mal, wie der funktioniert! Ich hab zunächst erzählt, wie die Grammatik aussehen muß, daß ich die <,=,> -Relationen festlege und dann beschrieben, wie das Parsen vor sich geht.
- Top-Down-Analyse: Muß die Grammatik bestimmten Bedingungen genügen? Ja, keine Linksrekursion, da sonst Endlosschleifen entstehen
- Und wenn da welche drin ist, lassen sie das Parsen dann sein? Nein, hier habe ich erzählt, daß man die Linksrekursion immer beseitigen kann, Verweis auf Greibach-NF (steht im 1820, hat ihn wohl etwas überrascht, daß das kam). Dann noch Linksfaktorisierung am Beispiel If-Statement erklärt.
- Wie erhalten sie die Steuermengen? Informal erklärt, saubere Definition sollte ich nicht mehr aufschreiben
- Wie implementieren sie daß? Direkt oder mit rekursivem Abstieg.
- Wie implementieren sie den rekursiven Abstieg genau? Prozedur für ein Nichtterminal in Pseudocode aufgeschrieben
- Wie kommen wir jetzt vom Parser zur Übersetzung? Attributierte Grammatik: Erweiterung der Symbole um Attribute und der Regeln um die Funktionen zur Attribut-Berechnung
- Was für Arten von Attributen kennen sie? Vererbt und Synthetisiert

- Was ist eine Syntaxgesteuerte Definition? Erlaubt Seiteneffekte wie z. B. Manipulation einer globalen Symboltabelle
- Was ist ein Übersetzungsschema? Habe etwas in der Form von $A \rightarrow B \{ \} C$ aufgeschrieben. Ihm war dabei wichtig, daß das Übersetzungsschema die Reihenfolge der Aktionen genau festlegt. (Ich hatte mich ehrlich gesagt beim Üben gefragt, wo eigentlich der Unterschied zwischen syntaxgesteuerter Definition und Übersetzungsschema ist).
- Was für spezielle Formen von attributierten Grammatiken kennen Sie? S- und L-attributierte Grammatik, habe dabei noch erwähnt, daß S-attribuiert besonders geeignet für Bottom-Up-Parser ist
- Wie kommen sie ausgerechnet auf diese komische Definition bei der L-attribuierten Grammatik? Das sind genau diejenigen Attribute, die ich mit rekursivem Abstieg = Tiefendurchlauf durch Syntaxbaum berechnen kann
- Wie genau implementieren sie eine L-attribuierte Grammatik im rekursiven Abstieg? Hier war ihm wichtig, daß die synthetisierten Attribute zu den Rückgabewerten und die vererbten Attribute zu den Eingabeparametern der Funktionen pro Nichtterminal werden.

Gesamteindruck:

Obwohl die Prüfung vom Ergebnis her problemlos verlaufen ist und Prof. Güting bei einigen kleinen Ungenauigkeiten auch drüber hinweggegangen ist, habe ich mich in der Prüfung ziemlich unwohl gefühlt. Ich bin in Prüfungen immer sehr nervös, aber z. B. bei Prof. Weihrauch oder Kamps ist das aber sonst in der Prüfung ziemlich verflogen. Hier wurde es dauernd schlimmer, obwohl ich eigentlich merkte, daß es gut läuft. Prof. Güting hat nicht einen Satz mit mir gewechselt, der nicht inhaltlich zur Prüfung gehört hätte.

Thematisch war ich überrascht, daß er relativ wenige Themen gefragt hat, diese dafür aber sehr stark im Detail. Ich hatte eher damit gerechnet, daß eher Top-Down- oder aber Bottom-Up-Analyse gefragt wird und dafür aber auf jeden Fall etwas aus KE5, insbesondere zur Speicherorganisation. Für die lexikalische Analyse hatte ich mich ziemlich darauf verlassen, daß das noch sitzt, weil ich a) erst im November mündliche Vordiplomprüfung in Theo hatte und b) im letzten halben Jahr beruflich ein wenig mit lex zu tun hatte, und hatte eigentlich nicht damit gerechnet, daß das in ernsthaftem Umfang dran kommt.

Das einzige, was mich ziemlich in Verlegenheit gebracht hätte, wären tiefergehende Fragen zu KE6. Ich halte es zwar für relativ unwahrscheinlich, daß er das prüft, aber ich würde es nicht riskieren, in die Prüfung zu gehen, ohne wenigstens ein paar Grundbegriffe daraus erläutern zu können.

Also, falls ihr euch entschließt, in die Prüfung zu gehen: Viel Erfolg und Glück dabei! Es gibt zwar vielleicht spannendere Themen und nettere Prüfer, aber ich finde den Kurs recht prüfungsfreundlich, weil das Thema recht kompakt und überschaubar ist. Er finde ihn größtenteils auch sehr verständlich geschrieben, insofern wird's schon klappen.

Prüfungsprotokoll

Diplomhauptprüfung Informatik
C1852 - Praktische Informatik Teil 2
1810 - Übersetzerbau
Prof. Dr. Güting
Dr. Erwig
ca. 25 min
18. März 1998

Fragen

- **Welche Aufgaben hat ein Übersetzer?**
- **In welchen Aufgabenbereichen werden Übersetzer eingesetzt?**
klassische Programmiersprachen, Datenbankabfragesprachen, Seiten- und Dokumentbeschreibungssprachen, Entwurfssprachen (VLSI) usw.
- **Nennen Sie die Eigenschaften und Unterschiede von Compilern und Interpretern!**
Arbeitsweise von Compiler und Interpreter erläutert, auf Nachfrage geschildert, wie man beide Techniken auch kombinieren kann (Beispiel: Java Virtual Machine)
- **Wie werden die zu übersetzenden Sprachen beschrieben?**
kontextfreie Grammatiken
- **In welche Phasen gliedert sich der Übersetzungsprozeß?**
Lexikalische Analyse, syntaktische Analyse, semantische Analyse, Zwischencodeerzeugung, Optimierung, Codeerzeugung (wichtig ist die Nennung der jeweiligen Ein- und Ausgaben sowie der wesentlichen Arbeitsschritte)
- **Welche grundsätzlichen Strategien gibt es für die Syntaxanalyse?**
Top-Down- und Bottom-Up-Analyse, Erläuterung der jeweiligen Vorgehensweisen
- **Welche einfachste Technik für Bottom-Up-Analyse wird im Kurs vorgestellt?**
Operator-Vorrang-Analyse, Erläuterung der Analysetechnik, Hinweis auf die Einschränkungen, denen die zugrunde liegende Grammatik genügen muß
- **Was versteht man unter attributierten Grammatiken?**
Definition attributierter Grammatiken, Unterschiede zwischen S-attributierten Definitionen (nur synthetisierte Attribute), allgemeinen syntaxgesteuerten Definitionen und L-attributierten Definitionen
- **Wie werden synthetisierte und vererbte Attribute während der Syntaxanalyse behandelt?**
- **Welche Aufgaben fallen bei der Verwaltung von Speicherplatz für Variablen an?**
- **Wie werden - u.U. geschachtelte - Deklarationen in verschiedenen Programmiersprachen behandelt?**
Hier erläuterte ich an Hand von Fortran, C und Pascal die unterschiedlichen Varianten; wichtig ist, zu erwähnen, daß die Schachtelungstiefe nur von der statischen Schachtelung des Quellcodes und nicht etwa von der Rekursionstiefe abhängig ist
- **Wie sieht das Speicherlayout im Detail aus?**
Layout schematisch dargestellt (Code, statischer Speicher, Stack und Heap)
- **Wie werden unterschiedliche Sichtbarkeitsbereiche implementiert?**
zwei Techniken: Prozedurrahmen, Displaymechanismus

Eindruck

Nach einer Vordiploms- und einer Hauptdiplomsprüfung war dies meine dritte Prüfung bei Prof. Güting. Die Prüfung verlief in einer sehr lockeren, angenehmen Atmosphäre. Prof. Güting stellte die

Fragen in recht allgemein gehaltener Form und ließ mich alles, was mir jeweils dazu einfiel, fast ohne Unterbrechungen erzählen. Die Fragen liefen im wesentlichen der Abfolge des Stoffs im Kurs entlang. Fragen zu Kurseinheit 6 (Übersetzung funktionaler Sprachen) wurden nicht gestellt; allerdings vermag ich nicht zu beurteilen, ob das die Regel ist oder eher ein Zufall war.

Aus meiner Sicht ist Prof. Güting als Prüfer uneingeschränkt zu empfehlen!

Viel Erfolg!