

Indermark, Theorie: CB, TES, SVVS

Prüfer: Indermark

Beisitzer: Noll

Prüfungsdatum: 7.4.1999

Vorlesungen:

- Compilerbau, WS 97/98
- Semantik und Verifikation verteilter Systeme, WS 98/99
- Termersetzungssysteme bei Baader, aus dem Skript (93/94) gelernt

Note: 1,3

Ich: Axel Simon, simona@pool.informatik.rwth-aachen.de

Indi ist dankenswerterweise mit Compilerbau angefangen und hat nach sehr langer Zeit mit der Bemerkung „Ich glaube, meine Uhr ist stehengeblieben“ noch kurz zu Termersetzungssystemen gewechselt, bevor dann der Studienberater mir meine Schwächen in SVVS unter die Nase gerieben hat. Ich fand die Prüfung angenehm, am Anfang hatte ich das Gefühl, die Zeit ginge nicht schnell genug um, da meine kurzen Antworten die angenehmen Themen so schnell vom Tisch fegten. Daraufhin habe ich versucht, etwas ausführlicher zu werden, was aber gar nicht gewünscht war. Bei den etwas schwierigeren Fragen habe ich dann von mir aus etwas länger gebraucht, was - neben meinen beiden Fehlern - zur Abwertung der Note führte.

Compilerbau (ca. 30 Minuten)

Indi: Aus welchen Teilen besteht ein Compiler?

Ich: Analyseteil (Frontend), Syntheseteil (Backend). Frontend besteht aus lexikalischer (Scanner), syntaktischer (Parser) und semantischer (Attributssysteme) Analyse. Backend aus Zwischencodegenerierung und einigem anderen.

Indi: Kommen wir zum Scanner. Was gibt der Scanner neben den Token aus?

Ich: Token können zusätzlich Attribute haben. Bei Identifiern oder Zahlen wird die Zeichenfolge bzw. die Zahl als synthetisches Attribut dem Parser übergeben.

Indi: Wie erzeugt man denn aus regulären Ausdrücken einen Scanner?

Ich: Kleenesche Konstruktion von regulären Ausdrücken (durfte die Konstruktion von α^* erklären). Vom NFA zum DFA durch Potenzmengenkonstruktion, ϵ -Closure erwähnt.

Indi: Wie sieht denn die Scannerkonstruktion in Form eines Programms aus?

Ich: war aufgeschmissen. Ich habe nicht gewagt zu sagen, daß ich das nicht kenne, da es in Compilerbau WS 98/99 neu dazu kam, da ich mir nicht so sicher war, ob es vielleicht doch an mir liegt.

Indi: Kommen wir zur LR Analyse. Was macht die GOTO Funktion, und was versteht man unter Shift und Reduce- Schritten?

Ich: Die LR Informationen auf dem Stack bestehen aus Regeln der Grammatik, in denen ein Punkt die schon eingelesenen Zeichen von den erwartenden trennt. Steht dieser Punkt bei einer Regel ganz rechts, wird diese Regel reduziert. Die GOTO Funktion sagt dann, wieviele Informationen vom Stack genommen werden müssen und in welchen Zustand man wechselt.

Indi: Was verstehen sie unter SLR(k)?

Ich: Das sind die schwächsten LR Grammatiken mit look-ahead. Um zu entscheiden, ob eine rechte Regelseite zu z.B. A reduziert wird, schaut man sich an, ob das nächste Eingabesymbol in $fo(A)$ ist.

Indi: Das mit dem fo -Aufruf kann man sich sparen, da man bei $A \rightarrow \gamma \cdot a$ sofort sieht, daß a auf der Eingabe sein muß; falls ein Nichtterminal dem Punkt folgt, werden diese Regeln ja rekursiv aufgenommen, so daß hinter dem Punkt immer ein Terminal steht. (Oder so ähnlich.)

Kommen wir zur semantischen Analyse. Was verstehen sie unter einer s -attributierten Grammatik?

Ich: Eine Grammatik ist s -attributiert, wenn nur synthetische Attribute benutzt werden. Dies steht im Gegensatz zu l -attributierten Grammatiken, die mit Einschränkungen auch inherite Attribute verarbeiten.

Indi: Was verstehen sie unter einem Abhängigkeitsgraphen?

Ich: Wenn man, ausgehend von einem konkreten Ableitungsbaum, für jede Gleichung einer Produktion von den Ausdrücken auf der rechten Seite zu den zu berechnenden eine Kante zieht.

Indi: Was bedeutet es, wenn ein Attributgleichungssystem Zykel hat? Wie kann man Zykel auch im allgemeinen feststellen?

Ich: Für den konkreten Ableitungsbaum bedeutet das, daß eine topologische Sortierung nicht möglich ist, man das Attributgleichungssystem also nicht auswerten kann. Im Drachenbuch steht ein Algorithmus, mit dem man für eine Grammatik überprüfen kann, ob sie zirkuläre Strukturen erzeugen kann. Wenn man z.B. die Abhängigkeitsgraphen der verschiedenen Produktionen übereinanderlegt, und es kommt ein zyklischer Graph heraus, dann ist das eine hinreichende Bedingung dafür, daß es konkrete Abhängigkeitsgraphen gibt, die zykel enthalten.

Indi: (Ist angetan von dem „hinreichend“ und erklärt, daß Kleene das in einer seiner früheren Arbeiten selber nicht erkannt hatte. Gut, dachte ich, dann wird er mich jetzt wohl nicht nach dem richtigen Algorithmus fragen. Und siehe da, er erklärte ihn selber.)

Indi: (schaut auf seine Uhr) Kommen wir noch kurz zur Codegenerierung. Wie wird Code für arithmetische Ausdrücke generiert?

Ich: (habe das aufgeschoben).

Indi: Welche zwei Möglichkeiten zur Übersetzung von booleschen Ausdrücken kennen Sie?

Ich: Jumping Code und Auswertung auf dem Stack.

Indi: Welche Besonderheiten gibt es beim Jumping-Code?

Ich: (War etwas ratlos und habe dann etwas über strikt, nicht strikt gesagt, darüber, daß Jumping-Code länger ist als der andere. Er wollte auf die Konstrukte hinaus, die es bei booleschen Ausdrücken so gibt (*if*, *while*) und wie man die Adresse bei der Codeerzeugung übergibt.)

Termersetzungssysteme (ca. 7,5 Minuten)

Indi: Ich glaube, meine Uhr ist stehengeblieben. Kommen wir zu Termersetzungssystemen von Herrn Baader. Können sie mir etwas über algebraische Spezifikationen sagen? Das müßte Herr Baader ganz am Anfang der Vorlesung erwähnt haben.

Ich: Man hat ein Universum was den Datentypen entspricht und definiert dann Operationen darauf.

Indi: Jaja, aber wie werden Eigenschaften von diesen Algebren angegeben?

Ich: Durch Gleichungen. Diese kann man dann wieder als TES auffassen. (Darauf wollte er also hinaus.)

Indi: Was besagt der Satz von Birkhoff?

Ich: (Da ich die Vorlesung nie gehört habe, war ich etwas verwirrt, als er den Namen englisch aussprach, denn ich hätte ihn beinahe nicht verstanden.) Der Satz ist relativ kurz, er besagt, daß $\xrightarrow{*}_E$ und $=^*_E$ die gleichen Relationen sind.

Indi: Jetzt möchte ich natürlich noch, daß sie mir diese Relationen genauer erklären. Sie syntaktische bitteschön.

Ich: (Überrascht, daß er nichts über freie Algebren, Kongruenzen und Quotientenalgebren wissen wollte, leitete ich recht langsam die Formel für die allgemeine Termersetzung her.)

Indi: Stellen sie daß doch mal graphisch dar. Als Baum bitteschön.

Ich: (habe das gerade noch hinbekommen, letztendlich hatte er das $|_w$ übersehen, daß ich an die linke Regelseite geschmiert hatte. Darauf wollte er hinaus.)

Indi: Wie sieht es denn mit dem Wortproblem aus?

Ich: Das ist in der Regel unentscheidbar. Es wird entscheidbar, wenn das TES konfluent und terminierend ist. Dann ermittelt man von den zu überprüfenden Wörtern die Normalformen und testet diese auf Gleichheit.

Indi: Was kann man denn machen, wenn ein TES diese Eigenschaften nicht besitzt.

Ich: Falls es nicht terminierend ist, kann man wohl nichts machen, da man unendlich absteigende Ketten nicht einfach abschneiden kann. Falls ein TES nicht konfluent ist, kann man aber durch umformen der Regeln oder durch hinzunahme (Vervollständigung) eventuelle die Konfluenz erreichen. (Davon kannte ich nicht mehr als die Kapitelüberschrift.)

Indi: Wegen der fortgeschrittenen Zeit, werde ich jetzt an Herrn Noll abgeben.

nicht
komplett

Gedächtnisprotokoll einer Diplomprüfung Theoretischer Informatik

Prüfer: Prof. Indermark (kurz Klaus)
Beisitzer: Olaf Chitil
Fächer: Compilerbau (WS 92/93) / Logikprogrammierung (WS 93/94)
Termerersetzungssysteme (WS 93/94 oder 95/96; gemäß Baader-Skript)
Datum: 15.01.96
Dauer: 50 min
Note: 1.0

Also für mich stand fest: Klaus beginnt mit Compilerbau, fragt - sagen wir - nach LR-Parsing und anschließend gehört die nächste halbe Stunde mir. Als Klaus das Wort 'Termerersetzungssysteme' über die Lippen ging, machte es 'Blupp' - ganz leise und es schien mir, als hätten Klaus und Olaf es nicht bemerkt - und die Seifenblase war zerplatzt. Man soll sich ja kein Bild machen...

Termerersetzungssysteme: (20 min)

Na gut, auch ein schönes Thema. Doch was gibt der Mann in seinem Einleitungsdialog alles von sich? Statt zu fragen, wie das Wortproblem oder der semantische bzw. syntaktische Folgerungsbegriff definiert sei, höre ich immer nur 'Kongruenzrelation' und 'Quotientenalgebra'. Jedenfalls spickte er seinen Monolog mit der Aussage, daß er auf den Satz von Birkhoff hinaus wolle. Artig nickend signalisiert man, daß man diesen Namen nicht mit einer Lebensmittelmarke verwechselt und folgert, daß er wohl mit der Kongruenzrelation die von E induzierte Gleichungstheorie meint. Also schnell Gültigkeit, Modell und Varietät erklärt, um dann \equiv_E definieren zu können. Einzig, daß reicht ihm nicht! Er will doch wissen, was eine Kongruenzrelation und eine Quotientenalgebra ist. Gut. Eine Kongruenzrelation ist eine Äquivalenzrelation (worauf Klaus sich zu dem Einwurf 'Das wissen wir' hingerissen sah. Ach, Klasse! Will er es nun wissen oder nicht?) die strukturverträglich ist. Bei der anschließend Definition der Quotientenalgebra legte er insbesondere Wert auf die Wohlfundiertheit, d.h. die Unabhängigkeit von der Wahl des Repräsentanten, was gerade aus der Kongruenzeigenschaft folgt. Den Beweis hierfür wollte er Gott sei Dank nicht haben, denn mein Auftreten war bis hierhin nicht gerade souverän (bzw. ich kam mir nicht so vor!), da ich mit seinen Fragen immer etwas anderes assoziiert habe bzw. nicht wußte, worauf er hinauswollte (daß $T(\Sigma, X)_{\equiv_E}$ die frei erzeugte Algebra über $V(E)$ mit Erzeugermenge X ist, wollte er gar nicht wissen). Ddabei ist die Frage, was eine Quotientenalgebra sei, gelinde gesagt, als legitim anzusehen! An die weiteren Fragen kann ich mich nun in genauerem Wortlaut erinnern. Beginnend mit einem klassischen, aber kurzen Blackout, den Klaus durchaus zu verzeihen gewillt ist, so man denn noch auf die Lösung kommt), lief's von da an in geregelten Bahnen.

Wie lautet die syntaktische Charakterisierung des Folgerungsbegriff?

Reduktionsrelation \rightarrow_{-E} angegeben (mit besagtem Aussetzer: Wie ging eigentlich eine Termerersetzung...? Irgendwann hatten wir es dann aber wieder!)

Was ist nun mit dem Wortproblem, daß durch diese Kongruenzrelationen gegeben ist?

I.a. unentscheidbar!

Können sie ein Beispiel hierfür angeben (Herr Baader hat doch sicher eins angegeben!)?

Was, ein Beispiel!? Ich erinnere mich da nur an ein unentscheidbares WES, das eine wilde Permutation des Alphabets darstellt. Ich habe ihm vorsichtshalber mal' ein nicht-konfluentes TES präsentiert, woraus aber meiner Ansicht nach noch nicht die Unentscheidbarkeit folgt (was ich auch Kund tat. Gleiches gilt auch für Nicht-Terminierung (z.B. kann man Wortproblem bzgl. Identitäten mit einer Kommutativitätsregel in Spezialfällen ggf. mit einem TES modulo dieser Regel lösen). Meine verzweifelten Blicke reichten offenbar, um endlich mal eine vernünftige Frage zu stellen.

Oder andersrum, wann ist das Wortproblem entscheidbar?

Das ist doch mal eine Frage, die man so richtig auskosten kann. Leider hat Klaus noch der Erläuterung der Bedingung 'Terminierung' und der 'Konfluenz' abgebrochen. Wichtig ist natürlich noch, daß man einen Reduktionsschritt effektiv bestimmen kann. Außerdem ist das Wortproblem auf Grundidentitäten entscheidbar. Jedenfalls konnte ich hier noch einiges über die Zusammenhänge von Konfluenz, lokaler Konfluenz und Church-Rosser anbringen.

- Beschreiben Sie Hyperquicksort.
Algorithmus erläutert.
- beschreiben sie das bitonische Sortieren.
Bitonisches Mischen/Sortieren erläutert
(Auch die nötigen Schrittzahlen und die Komparatorzahlen sollte ich angeben)
- Gibt es eine Implementierung dieses theoretischen Prinzips auf einem realen Rechner ?
Bitonisches Mischen/Sortieren auf Perfect-Shuffle-Topologie.
- Neues Thema : Wie geht man vor beim implementieren der parallelen Präfixe auf einem HC ?
Kostens optimaler PRAM-Algorithmus als Grundlage, in zweiter Komponente alle Elemente von Prozessoren mit kleinerer Nummer addieren.
- Wenn man die HC-Topologie betrachtet, was ist dann nicht so gut ?
Kein konstanter Kantengrad (Verbesserung : Cube-Connected-Cycles).

2. Effiziente Algorithmen

- Beschreiben sie den Strassen-Algorithmus.
Algorithmus beschrieben, Aufwand angegeben.
- Beschreiben sie die Heuristik für das TSP-Problem.
Gegeben ist ein Graph, in dem die Dreiecks-Ungleichung gilt. Idee : Bestimme minimalen spannenden Baum und suche „Abkürzungen“.
- Effiziente Bestimmung des MST ?
Kruskal-Algorithmus mit Zykelprüfung über UNION-FIND.
- Was ist die Idee der FFT ?
Auswertung eines Polynoms an den n-ten Einheitswurzeln.
Habe dann ausgeholt und die Einheitswurzel-Eigenschaften aufgezählt sowie deren Verwendung bei der Implementierung der FFT.

3. Termersetzungssysteme

(Hier wurden mir nur *sehr* allgemeine und wenige Fragen gestellt)

- Wie lautet die Konfluenzbedingung ?
Definition angegeben.
- Erläutern sie die Bildung von kritischen Paaren.
Für TES erläutert, welche Situationen (bei terminierenden TES) kritisch sind. Definition der kritischen Paare angegeben.
- Was ist die Idee der Vervollständigung ?
Man hat eine Menge von Identitäten gegeben, zu der man ein äquivalentes, kanonisches und endliches TES sucht. Einfachen Vervollständigungs-Algorithmus angegeben.

- Dieses ist nun die semantische Ebene, was ist denn mit der syntaktischen Ebene ?
Die syntaktische Ebene bildet die Reduktionsrelation \rightarrow_E (welche ich dann definieren sollte). Ferner sollte ich den Satz von Birkhoff erklären (siehe Skript).

Die Prüfungsatmosphäre war sehr angenehm und Prof. Indermark versteht es, gute Hilfestellungen zu geben. Unmittelbar nach der Prüfung rechnete ich mit einer doch etwas schlechteren Note, da ich viele Dinge nicht auf Anhieb wußte. Anscheinend gefällt es Prof. Indermark aber recht gut, wenn man nach Hilfestellungen die gewünschte Antwort herleiten kann, man sollte sich aber auf jeden Fall für einige Standardfragen die Antwort (genau formuliert) vor der Prüfung überlegen.

- *Transformation eines funktionalen Programms in ein Logikprogramm*
Bei der Transformation werden n -stellige Funktionen durch $(n+1)$ -stellige Relationen (ihre Graphen) dargestellt. Weiterhin werden die natürlichen Zahlen durch die 0 und ein einstelliges Funktionssymbol s codiert.
- *Definition und Diskussion der Fixpunktsemantik von Logikprogrammen*
Hier sollte ich unter anderem die ersten zwei Schritte der Fixpunktiteration beschreiben: Im ersten Schritt enthält die Semantik alle Grundinstanzen der Tatsachenklauseln des Logikprogramms und im zweiten Schritt kommen die Grundinstanzen der linken Regelseiten der Programmklauseln hinzu.
- *Was ist der Cut und was bewirkt er?*
Erklärung der Wirkungsweise des Cuts anhand eines Beispiels. Einsatz des Cuts zur Simulation von if-then-else. Hier kam auch eine Diskussion der Close World Assumption auf und wie der Cut zur Realisation der negation as finite failure eingesetzt werden kann.
- *Wie sieht ein SLD-Baum aus?*
Definition des SLD-Baums und Beschreibung, was an den „Enden“ des Baums passiert (fail, success oder unendliche Berechnung).
- *Universalität von PROLOG*
Hier haben wir die primitive Rekursion behandelt.

Termersetzung:

- *lokale Konfluenz von Termersetzungssystemen*
An welchen Stellen kann die Reduktion stattfinden, wenn ein Term mehrere Nachfolger besitzt. Was gibt es hier für kritische Situationen? Schließlich Definition der Menge der kritischen Paare zwischen zwei Regeln eines Termersetzungssystems.
- Kurzer Exkurs zur *Knuth-Bendix-Vervollständigung*.
- *Wie hängen lokale Konfluenz und Konfluenz zusammen?*
Ist das betrachtete Termersetzungssystem terminierend, so folgt aus der lokalen Konfluenz die Konfluenz.

Prüfung	Theoretische Informatik Prof. Dr. F. Baader	
Themen	Termersetzungssysteme Logische Methoden der Wissensrepräsentation Automatentheorie	
Termin	22. April 1996	NOTE: 4.0

Logische Methoden der Wissensrepräsentation

- Was ist der Unterschied zwischen einem Datenbanksystem und einem Wissensbasierten System?
- Konzept- und Rollenkonstruktoren (Syntax und Semantik von Konzeptbeschreibungen), Boole'sche Operationen; Größte-Kleinste-Konzept; Rollenrestriktionen, Zahlenrestriktionen; Wertbereichseinschränkung; Komposition; Rollenfüllerübereinstimmung; Konjunktion;
 $(r_1 \sqcap r_2)^s := \{ (d, e) \in D_1 \times D_2 \mid (d, e) \in r^s \text{ und } e \in C^s \}$ und
 $(r_1 \sqsubseteq r_2)^s := \{ \phi \in D_1 \mid r_1^s(\phi) \subseteq r_2^s(\phi) \}$ erklären.
- Schlußfolgerungsmechanismen erklären:
 Subsumption, $C \sqsubseteq D$ gdw. $C^s \sqsubseteq D^s \forall$ Interpretationen S .
 Terminologisches und assertionales Schließen.
- Was bedeutet „Endliche Modelleigenschaft“ ?

Termersetzungssysteme

- Ist Terminierung entscheidbar ?
 Im allgemeinen nicht, sondern nur für spezielle Systeme.
- Reduktionsordnungen und Vereinfachungsordnungen erklären ?
 Wie zeigt man Noetherscheit bei Vereinfachungsordnungen ?
- Konfluenz beschreiben. Bedeutung der Lokalen Konfluenz.
 Wie zeigt man, daß bei gegebener Terminierung lokale Konfluenz reicht, um Konfluenz zu zeigen ? (Noethersche Induktion; Beweiskizze)
- Bedeutung der Kritischen Paare und wie werden diese gebildet ?
 (Unterschied: Grundterme und Terme mit Variablen)

Automatentheorie

- Was ist ein Büchi-Automat ?
 Unendliche Sprachen, Akzeptanzbedingung.
- Abschlußeigenschaften ? (Durchschnitt, Vereinigung Komplement)
 Wie zeigt man Abschluß unter Durchschnitt ? (Entsprechenden Automaten angeben)
 Komplement: Unterschied zwischen deterministischen und nichtdeterministischen Büchi-Automaten ?
- Zusammenhang zu logischen Formeln ?
 SIS beschreiben: Geschlossene PL1-Formeln mit \wedge, \vee, P_n, Q_n und Nachfolgerfunktion s , sonst keine weiteren Funktionen.
- Was ist ein Baumautomat ?
 BW-Baumautomat und WB-Baumautomat beschreiben.
 Übergangsrelation für WB-Baumautomat definieren: $\Delta_a : Q \rightarrow 2^{Q^+}$ für $a \in \Sigma_n$ für $n > 0$.
 Lauf: $I: \text{dom}(t) \rightarrow Q$ mit $(I(u_0), \dots, I(u_{n-1})) \in \Delta_a(I(u))$ für $t(u) = a \in \Sigma_n$.
- Welche Sprachen werden von welchen Baumautomaten akzeptiert ?
 ω -Baumsprachen definieren. Nichtdet. und det. BW-Baumautomat und nichtdet. WB-Baumautomat gleich, det. WB-Baumautomat erkennt nur Teilmenge.



- Lösen von Gleichungen in $\mathcal{T}(\Sigma, X)/\equiv_E$.
- Anwendung: z.B. Termersetzung modulo einer Gleichungstheorie.
- Welche Unifikationstypen gibt es?
 - unitär, finitär, infinitär, Typ 0
- Skizzieren sie die Idee der C-Unifikation.
 - \equiv_C -Klassen der zu unifizierenden Terme bestimmen und für alle Kombinationen den syntaktischen mgu berechnen.
 - Herr Baader hat nie nach kompletten Algorithmen gefragt, entscheidend war immer die Grundidee.
- Warum ist A-Unifikation infinitär?
 - Beispiel mit der zugehörigen minimalen vollständigen Menge angeben.
- Warum ist AI-Unifikation vom Typ 0?
 - Beispiel und Hilfsbehauptung angeben. Anschließend gezeigt, daß aus der Hilfsbehauptung die Nicht-Existenz einer minimalen vollständigen Menge für das Beispiel folgt. (siehe Vorlesung)

2.2 Termersetzungssysteme

- Was ist das Wortproblem?
 - $s \equiv_E t?$
- Ist das Wortproblem entscheidbar?
 - I.a. nicht. Jedoch falls ein äquivalentes kanonisches TES existiert. Außerdem für endliche Mengen von Grundidentitäten.
- Ist die Terminierung von TES entscheidbar? Warum nicht?
 - Simulation einer Turingmaschine durch ein WES, Simulation eines WES durch ein TES.
 - Wichtig ist auch, daß zu jedem Wort u , von dem eine unendliche Reduktionskette ausgeht, ein Berechnungszustand der TM existiert, so daß die TM nicht terminiert. (Beweisidee)
- In welchen Fällen ist die Terminierung von TES entscheidbar?
 - Für Grundtermersetzungssysteme
 - Falls eine mit dem TES verträgliche Reduktionsordnung existiert.

- Was ist eine Reduktionsordnung, worin unterscheidet sie sich von einer Vereinfachungsordnung?
 - Definitionen angeben.
 - Vereinfachungsordnungen besitzen die Untertermigkeit.
- Warum ist jede Vereinfachungsordnung auch eine Reduktionsordnung?
 - Man betrachtet die homöomorphe Einbettung und zeigt, daß es bezüglich dieser nur gute Folgen gibt... Hier war auch nur nach den wesentlichen Ideen des Beweises gefragt.
- Zeigen sie die Terminierung von

$$\rightarrow \subseteq \mathbb{N}^* \times \mathbb{N}^*$$

$$u(i+1)v \rightarrow iuvi$$

- Zeigt man mit Multimengenordnung.
- Warum ist die Multimengenordnung Noethersch, wenn die induzierende Ordnung Noethersch ist?
 - Beweisskizze: Konstruktion einer Folge von Bäumen, Lemma von König.
- Wann ist Konfluenz von TES entscheidbar? Warum?
 - Bei terminierenden TES ist Konfluenz entscheidbar. Kritische Situationen aufgezählt und gezeigt, warum es nur endlich viele sind.
- Nennen sie eine Situation, in der lokale Konfluenz aber keine Konfluenz vorliegt.
 - $a \leftarrow b \longleftrightarrow c \rightarrow d$.
- Wozu braucht man Termersetzung modulo Gleichungstheorien?
 - Es gibt Fälle, in denen die Vervollständigung ohne Abbruch nicht terminiert.
- Erläutern sie die drei Varianten $(R, R, A, R/A)$. Warum dürfen bei Reduktion mit R nur linkslineare Regeln verwendet werden?
 - Beispiel angeben.

Fach: Theoretische Informatik
Prüfer: Prof. Dr. Indermark
Datum: Oktober 1996
Gebiete: Funktionale Programmierung (FP)
Termersetzungssysteme (TES)
Semantik von Progsprachen (SEM)
Echtzeitsysteme (EZS)
Note: 1.7

FP:

- kurzer Monolog über den Anfang der Vorlesung, dann
- ? Wie sieht Stackmaschine für TermAuswertung aus?
- ! 3 Stacks, Befehle nur EXEC und LOAD.
- ? Wie sieht die Übersetzung aus?
- ! trans und et für Terme aufgeschrieben.
- ? Gut, nun zu rekursiven Funktionsdefinitionen:
Wie ist deren Reduktionssemantik definiert?
- ! über Reduktionsrelation \Rightarrow . Zunächst Red-Regeln \rightarrow erklärt und aufgeschrieben, dann allgemeine \Rightarrow aufgeschrieben, erklärt, daß $u \Rightarrow u$ der Vollständigkeit halber (für versch. tief geschachtelte Terme) vorhanden sein muß. Deshalb aber nicht deterministisch ...
- ? Schlimm?
- ! Nein, da bei Ex. eindeutige Normalform, denn \Rightarrow ist Church-Rosser.
- ? \Rightarrow_{LO} ?
- ! definiert nach Vorlesung.
- ? endrekursive Funktionen, warum "endrekursiv"?
- ! Rekursion weder geschachtelt noch in Grundoperationen.
- ? was ist mit cbv und cbn?
- ! sind hier gleich.
- ? gleich?
- ! die Strategien sind natürlich nicht gleich, aber ihr Ergebnis.
- ? Aha, in welchem Verhältnis stehen denn cbv und cbn zueinander?
- ! $cbv \leq cbn$ bzgl. Graphinklusion.
- ? Kommen wir nun zum Lambda Kalkül.
Wie ist denn die beta-Reduktion definiert?
- Leider ging genau in diesem Moment die Sonne unter, und es breitete sich selbst in Indy's Büro eine unglaublich tiefe, dunkle Nacht aus:
Wer war das: Beta? getypt, ja! Ich begann irgendwie zu schrumpfen, die nächsten 5 Minuten möchte ich meiner Leserschaft lieber vorenthalten.
Mit sehr viel Hilfe habe ich es am Ende doch noch aufschreiben können, die Schlacht war allerdings verloren.
- gut 25 Minuten waren nun um, neues Thema:

TES:

- ein viertelständiges Frage- und Antwortspiel über die algebraischen Grundlagen. Praktisch ein Wort-für-Wort-Abfragen der Seiten 47-50 im Baader Skript (1995) mit Modell, semant. Folgerung, $=_E$, Kongr.-Rel., Quotientenalgebra, dann Satz von Birkhoff und \rightarrow_E definiert.
- alles sehr schön, die Sonne loderte schon längst wieder am Firmament.
- Huch, es wird Zeit:

SEM:

? in der Vorlesung, wie wurde denn da die denotationelle Semantik von WHILE erklärt?

! Ich fing mit induktivem Aufbau an ..., aber er brach gleich ab und wollte nur die Semantik des while-Befehls genau hören: Fixpunkt des Einsetzungsfunktional!

? Aha, warum ist denn das so?

! Dank Tarski.

? Wie lautet der Fixpunktsatz?

! FR vollst. HO, Φ stetig \rightarrow Φ besitzt kleinsten (in der Hektik vergessen)

FP, nämlich Supremum der Φ^i (bottom).

? Wie viele FP gibt es? (wie gesagt: "kleinster" vergessen!)

! einen ... ääh, natürlich beliebig viele, aber nur einen Kleinsten!

(Oh Gott, doppelt doof!)

- kurze Einführung über parallele Bearbeitung und Guarded Commands.

? Wie ist die Semantik für kommunizierende Prozesse definiert?

! SOS induktiv, das wichtigste dabei ist die Interleaving-Semantik, d.h. Prozesse dürfen synchronisiert ablaufen. Erzwungen wird dies durch die Restriktion.

- sehr gut, letztes Thema:

EZS:

? kontinuierliche Semantik, was können sie dazu sagen?

! Auswertungsgraph ist Semantik eines Zeitgraphen und kann diskret oder kontinuierlich ausgewertet werden ...

? Ja, aber hat nicht Einstein gesagt, daß alles relativ ist und ...

! Ich denke Sie wollen auf driftende Uhren hinaus ...

? ja, genau.

! Die sind nötig, da die Welt richtigerweise nicht richtig tickt. In der Theorie jedoch nicht weiter relevant, da sich Zeitgraphen mit beliebigem Drift $[l,u]$ zu K-Zeitgraphen umformen lassen (multiplizieren aller unteren Vergleiche mit u , aller oberen mit l) und danach mit Division durch kgV K in klassische 1-ZG.

? Angenommen, wir hätten hier zwei exakte Atomuhren und eine lassen wir zum Mond und wieder zurück fliegen, zeigen beide nachher noch die gleiche Zeit?

! Nein.

? Richtig, denn ...(weiß nicht mehr genau ?!)

- Dankeschön, das wars: 1.7

- Ich weiß leider nicht mehr wann, aber ich bekam auch die Frage "welche rekursiven Funktionen sind total?", wußte jedoch keinerlei Antwort darauf. (es wären gewesen: die primitiv-rekursiven Funktionen)

Begründung:

unverzeilich waren beta-Reduktion und kleinster FP. "Und wer eine eins haben möchte sollte Fragen aus dem Vordiplom beantworten können, ansonsten echt gut".

Toller Tip:

seid nicht so hektisch wie ich! Mann und Frau hat genug Zeit, um auf alle Fragen in Ruhe zu antworten.

Ansonsten:

Atmosphäre echt fair, aber es gibt nichts geschenkt.

Gedächtnisprotokoll

Theoretische Informatik

1 Allgemeines

- Prüfer: Prof. Indermark (im folgenden KI)
- Beisitzer: Volker Stolz
- Prüfungsgebiete:
 - Compilerbau (WS 00/01) 4 SWS
 - Funktionale Programmierung Kap. 1-6 (SS 99) 2 SWS
 - Termersetzungssysteme (WS 00/01) 4 SWS
 - Automata and Model-Checking (WS 00/01) 2 SWS
- Datum: 23. August 2001
- Uhrzeit: 9.00 Uhr (typische Indermark-Zeit)
- Dauer: 45 Minuten
- Note: 1,0
- Name: Rajendra Persaud

2 Prüfungsverlauf

Es ging praktisch ohne Umschweife los: "Morgen die Herren. Herr Stolz, Sie sitzen da, Herr Persaud, Sie sitzen da und ich sitze hier." Dann hat KI noch kurz die Gebiete erwähnt, über die die Prüfung gehen würde und meinte, daß er gerne mit Model-Checking anfangen wollte. Fand ich nicht so toll, da es dazu noch kein Prüfungsprotokoll gab und ich daher noch keine Fragen kannte. Aber gut ...

2.1 Automata and Model-Checking (10 Minuten)

- Ich möchte mal mit Model-Checking beginnen. Da ist von Strukturen die Rede. Welche Strukturen sind gemeint?
Das sind Transitionssysteme, die sogenannten Kripke-Strukturen.

- Genau. Können Sie die mal beschreiben?

Eine Kripke-Struktur ist ein Tupel $M = (S, R, L)$ mit endlicher Zustandsmenge S , einer Übergangsrelation R und einer Beschriftungsfunktion $L : S \rightarrow 2^{\{p_1, \dots, p_n\}}$ über den atomaren Formeln p_1, \dots, p_n .

- Neben den Kripke-Strukturen gab es auch noch Logiken. Welche?

Im wesentlichen haben wir uns mit LTL und CTL beschäftigt.

- Können Sie die Syntax der until-Formel in LTL angeben?

$$\varphi = \psi_1 \text{ U } \psi_2$$

- Und wie ist die Semantik definiert?

Die Semantik ist über Beschriftungspfade $\alpha \in (2^{\{p_1, \dots, p_n\}})^\omega$ definiert:

$$\alpha \models \psi_1 \text{ U } \psi_2 \text{ gdw } \exists j \alpha^j \models \psi_2 \wedge \forall i \ 0 \leq i < j \rightarrow \alpha^i \models \psi_1$$

- Und wie hängt das nun mit der Kripke-Struktur zusammen?

Eine Kripke-Struktur M mit Startzustand s ist Modell einer LTL-Formel φ gdw $(M, s) \models A\varphi$ in CTL*, d.h. φ muß auf allen Beschriftungspfaden α von s aus gelten.

- Und wie funktioniert nun LTL-Model-Checking?

Ich erzähle, daß man die Kripke-Struktur in einen Büchi-Automaten umwandelt, dann zu der Formel φ zunächst einen generalisierten und dann auch einen normalen Büchi-Automaten konstruiert, den Produktautomaten bestimmt und darauf den Leerheitstest durchführt.

- Ist der Leerheitstest entscheidbar?

Ja.

- Und wie führt man den durch?

Man sucht einen Endzustand q , der vom Startzustand q_0 aus erreichbar ist, und von dem aus ein nicht-leerer Pfad zu sich selbst existiert. Ich habe bei der Erläuterung das entsprechende Bild dazu aufgemalt.

- Was ist denn für die Entscheidbarkeit unabdingbare Voraussetzung?

Die Endlichkeit des Zustandsraums.

- Wie funktioniert denn die Konstruktion des Automaten \mathfrak{A}_φ aus der LTL-Formel φ ?

Ich muß grinsen, da ich mich aufgrund langem Unverständnis mehrfach und ausführlich mit dieser Konstruktion befaßt hatte, aber mir doch nicht vorstellen konnte, daß KI mich drauf ansprechen würde. Hat er aber doch,

und deshalb mußte ich grinsen, was KI zu dem Einwurf hinriß: "Herr Persaud, wieso grinsen Sie?". Da hab ich ihm gesagt, daß die Konstruktion so aufwendig sei, worauf er meinte, "ja, ja, deshalb würde er ja auch drauf eingehen!".

Ich erkläre also zunächst den Begriff der φ -Erweiterung und sage, daß man eine korrekte φ -Erweiterung durch 6 Bedingungen charakterisieren kann und daß der Automat \mathfrak{A}_φ dann so konstruiert wird, daß er diese 6 Bedingungen erfüllt.

- Woraus besteht denn der Zustandsraum von \mathfrak{A}_φ ? *Die genaue Konstruktion interessierte ihn wohl nicht so sehr.*

Jeder Zustand ist ein Wahrheitswertetupel aus der Menge $\{0, 1\}^{n+m}$, so daß ein Lauf durch den Automaten eine φ -Erweiterung ist.

2.2 Funktionale Programmierung (10 Minuten)

- Kommen wir zur funktionalen Programmierung. Wir haben da ja die Reduktionssemantik von rekursiven Funktionsdefinitionen betrachtet. Können Sie die wesentlichen Begriffe dabei erläutern?

Die Reduktionssemantik ist eine operationelle Berechnungssemantik und basiert auf den Berechnungstermen $BT := T_{\Sigma[\bar{F}]}(A)$. KI ergänzt, daß sie keine Variablen und kein \perp enthalten.

- Dann gibt es noch die Berechnungsregeln. Welche sind das?

Grundreduktion, Verzweigungsreduktion und F-Reduktion. Die F-Reduktion sollte ich dann konkret aufschreiben.

- Und wie sieht es mit der Reduktionsrelation aus?

Habe ich nach Vorlesung definiert.

- Wie reduziert man denn bei der Regel $\varphi u_1 \dots u_n \Rightarrow \varphi v_1 \dots v_n$?

Im Prinzip immer parallel, wozu er meint "Na ja, nicht immer.", worauf ich aber ergänze, entweder mit leeren oder mit echten Reduktionsschritten. "Na ja, so gesehen hätte ich wohl recht."

- Die Reduktionsrelation ist ja konfluent (*er wollte bei der vorhergehenden Frage wohl darauf hinaus, daß man nicht-deterministisch reduziert*). KI guckt mich fragend an, ich gucke fragend zurück, wir schweigen uns an. Ich denke, wenn er will, daß ich Konfluenz erläutere, dann kann er das doch sagen. Schließlich fängt er an zu grinsen und nötigt mich dann doch dazu, ohne was gesagt zu haben.

Wenn es eine Normalform gibt, so ist diese eindeutig.

- Was wäre hier eine solche Normalform?

Ein Element aus A .

- Wie zeigt man denn die Konfluenz?

Sage, daß *er* Konfluenz der Reduktionssemantik gezeigt hätte, indem man die Fixpunktsemantik auf die Berechnungsterme überträgt und dann Invarianz bei den Reduktionsschritten nachweist. Ich füge hinzu, daß man mit den Methoden der Termersetzung wohl auch Schwierigkeiten hätte, da die Reduktionsrelation ja nicht-terminierende Reduktionen enthalten kann. Wir diskutieren kurz darüber, bis er meint, daß man dann praktisch fordern müßte, daß es stets wenigstens einen echten Reduktionsschritt gibt.

- Wie ist denn nun die Reduktionssemantik definiert?

$Red[[R, \sigma]] : A^w \rightarrow A^s$ mit $(a_1, \dots, a_n) \mapsto a : \Leftrightarrow F_1 a_1 \dots a_n \xrightarrow{*} a$

- Wozu benötigt man den Striktheitsindex σ ?

Häh? Na ja, ich sage, um strikte und nicht-strikte Funktionen zu bekommen. *War aber wohl nicht wirklich das, was er hören wollte.*

- Was bedeutet denn Striktheit?

Sage, daß der Begriff nur bei flachen Halbordnungen Sinn macht und gebe dann die Definition von strikten Funktionen an.

- Können Sie sich z.B. bei den booleschen Funktionen solche vorstellen, die nicht-strikt sind?

* Das *and* zum Beispiel.

- Wie sieht denn das strikte bzw. nicht-strikte *and* aus?

Erwähne, daß es beim nicht-strikten *and* das links-sequentielle, das rechts-sequentielle und das parallele gibt und erläutere alle Varianten an der Wertetabelle.

- Haben Sie eine Vorstellung davon, wie man das parallele *and* implementiert?

Darüber hatte ich nur in Prüfungsprotokollen gelesen und es erst später einigermaßen verstanden. Bei der Implementierung des parallelen *and* müssen die beiden Teilausdrücke et_1 und et_2 von et_1 *and* et_2 stückweise abwechselnd ausgewertet werden. Sobald dann einer zu false ausgewertet, ist das Gesamtergebnis eben auch false.

2.3 Termersetzungssysteme (10 Minuten)

- Das soll genügen, gehen wir zu Termersetzungssystemen. Sie haben da zu Anfang über algebraische Spezifikationen gesprochen. Wissen Sie, was das ist?

Als ich auf diese Frage in einem Prüfungsprotokoll gestoßen war, hatte ich gedacht, häh, wann haben wir denn das gemacht, aber beim Nachgucken stellte sich dann heraus, daß wir es in der Tag behandelt hatten. Dient KI wohl als Einstiegsfrage, um auf Identitäten zu kommen.

Ich sage also, daß wir da Funktionssymbole mit Hilfe von Identitäten definiert haben. KI stört sich etwas an “definiert”, erklärt auch was dazu, was ich aber nicht verstehe. Ich erwähne noch “+” als Beispiel, was ich dann auch mal aufschreiben sollte.

- Gut, Sie haben dann auf syntaktischer und semantischer Ebene die Äquivalenz von Termen behandelt. Erläutern Sie mir das bitte und auch den bestehenden Zusammenhang!

Ich definiere die syntaktische Reduktionsrelation \rightarrow_E für eine Menge von Identitäten E und male auch das entsprechende Bäumchenbild dazu. Danach hätte er ja eh gefragt.

- *Jetzt kommt die erste wirklich böse Frage. Ich hatte auf meinem Blatt nämlich im oberen Bereich noch was zur Reduktionsrelation in FP stehen und unten ja gerade \rightarrow_E in TES definiert.*

Warten Sie mal, wie wäre das denn, wenn wir das hier auf die Reduktionsrelation von da oben übertragen. Wir würden dann die Termpaare, die aufgrund von Reduktionsregeln in der Relation stehen, als Identitäten E auffassen und dann die Reduktionsrelation \rightarrow_E definieren. Wie sieht das aus?

Äh, kurzer Schock, was will er eigentlich? Nachgefragt, was er sich konkret vorstellt. Als er es wiederholt hat, kommt mir immer noch keine zündende Idee. Ich meine lediglich, daß ich spontan sagen würde, daß man die Reduktionsrelation \Rightarrow aus FP mit \rightarrow_E nachspielen könne.

- Ja, aber ... *er erzählt irgendwas, woraus sich zum einen schließen läßt, daß diese Frage nicht vorbereitet gewesen ist, sondern sich völlig spontan bei ihm ergeben hat. Deswegen war er sich selbst nämlich auch nicht sicher, wie es denn nun genau ist. Er meinte auf jeden Fall, daß sich das überschneiden müßte.*

Dazu meine ich dann aber wieder, daß es doch so gehen müßte, wie er es sich vorgestellt hat, man hätte dann zwar keine Variablen in den Termen mehr und daher wäre auch keine Substitution mehr nötig für \rightarrow_E , aber es müßte doch gehen.

- *War damit zufrieden und ging zur semantischen Ebene über. Wie ist denn da die entsprechende Relation definiert?*

Ich sage, daß da mehrere Begriffe eine Rolle spielen und erkläre Gültigkeit, Modell, Varietät, semantische Konsequenz und Gleichungstheorie.

- So, und jetzt den Zusammenhang bitte!

Der ergibt sich aus dem Satz von Birkhoff. Ich sage, daß danach zwei Terme genau dann syntaktisch äquivalent sind, wenn sie auch semantisch äquivalent sind.

2.4 Compilerbau (15 Minuten)

- Gut, ich denke, das genügt. Kommen wir zum Compilerbau. Zum Parser. Wie schreibt man einen Parser? *Ich wollte schon anfangen, den recursive descent parser zu erläutern, als er meinte: Nein, zum Scanner. Wie funktioniert der? Ja, was denn jetzt?*

Ich sage, daß der auf regulären Ausdrücken und endlichen Automaten basiert. Die regulären Ausdrücke beschreiben Symbolklassen und werden durch die Thompson-Konstruktion in NFAs und durch die Potenzmengenkonstruktion in DFAs umgewandelt.

- Da haken wir mal ein. Bei der Umwandlung in NFAs. Wir wollen das nun noch etwas erweitern. Und zwar um den Schnitt. Wie könnte man das machen?

Bin etwas verwirrt. Schnitt? Bei regulären Ausdrücken? Was für eine wilde Konstruktion soll das denn geben? Sage also nichts zu regulären Ausdrücken, sage, daß ich es bei DFAs auf die übliche Art machen würde mit Schnittbildung der beiden Endzustandsmengen.

- Welche Invarianten hält man denn bei der Thompson-Konstruktion ein?

Man sorgt dafür, daß jeder konstruierte Automat genau 1 Anfangs- und genau 1 Endzustand hat. Darüber hinaus muß der Anfangszustand Quelle und der Endzustand Senke sein.

- Ja genau. Und jetzt überlegen Sie doch mal! *Als sich bei mir nichts tut, sagt er "Holzhammermethode".*

Na ja, wenn Sie Holzhammermethode sagen, dann würde ich die regulären Ausdrücke in einen DFA umwandeln und dann den Schnitt bilden.

- Ja, und worauf müssen Sie dann noch achten?

Ach ja, daß die Invarianten eingehalten werden.

- Und das heißt?

Daß der Startzustand Quelle ist ... aber das ist er doch in jedem Fall ... ach nein, man nimmt dann einen neuen Startzustand und fügt eine ϵ -Transition von diesem zu dem alten Startzustand hinzu. Und genauso nimmt man einen neuen Endzustand und fügt für den alten Endzustand (KI wirft sofort ein, daß es i.a. auch mehrere geben kann), und fügt also für die alten Endzustände je eine ϵ -Transition zu dem neuen Endzustand hinzu.

- Gut, nun zum Parser. Wie funktioniert für eine beliebige kontextfreie Grammatik der LL(1)-Test?

Man berechnet die la -Mengen und prüft diese für Regelalternativen auf Disjunktheit.

- Wie sind denn die la -Mengen definiert?

Für $\pi = A \rightarrow \alpha$ ist $la(\pi) = fi(\alpha fo(A))$.

- Wie sieht so eine Menge aus?

$la(\pi) \subseteq \Sigma_\epsilon$

- Wann ist $\epsilon \in la(\pi)$?

Wenn $\alpha \xrightarrow{*} \epsilon$ und $\epsilon \in fo(A)$.

- Nun zu LR(0)-Mengen. Wie kann man diese berechnen?

Sage, daß die Berechnung auf der Potenzmengenkonstruktion eines NFA basiert. Er stößt sich ein wenig an "basiert" und erklärt wieso. Nun ja, ich habe dann angefangen, die Konstruktion zu erläutern und sage, daß die Zustandsmenge die Menge aller LR(0)-Auskünfte ist.

- Wie sieht denn die Übergangsrelation aus?

Zum einen $[A \rightarrow \beta_1 \cdot X \beta_2] \xrightarrow{X} [A \rightarrow \beta_1 X \cdot \beta_2]$ und zum anderen $[A \rightarrow \beta_1 \cdot B \beta_2] \xrightarrow{\epsilon} [B \rightarrow \cdot \delta]$, falls $B \rightarrow \delta$ eine Regel der Grammatik ist.

- Nun, Sie könnten mir sicherlich noch viel erzählen, aber wir wollen es dann hiermit bewenden lassen. Wenn Sie bitte kurz draußen warten, damit ich mich mit meinem Assistenten beraten kann.

3 Bemerkungen

Also, wie eigentlich überall zu lesen gewesen ist, war die Atmosphäre ruhig und angenehm, teilweise ja auch witzig, also auf keinen Fall angespannt oder so. Der Assi hat bei mir keinen Ton gesagt, sondern nur protokolliert. Mir schien, daß KI oft eine Einleitungs-Frage stellt, zu der man dann eine halbe Stunde was erzählen

könnte. Man muß aber nur einfach irgendwo anfangen, und er fragt dann weiter, eben kleinschrittiger.

Außerdem will er einen mit keiner Frage irgendwie hereinlegen oder linken, alle Fragen sind ernsthaft und vernünftig. Ich hatte z.B. bei der Betrachtung des Schnitts bei regulären Ausdrücken zuerst an eine Finte oder so was gedacht, das aber nicht ausgesprochen. Hat sich ja dann auch nicht als eine solche herausgestellt. Ich denke, das wäre nicht sein Stil.

Am Ende war ich eher überrascht, daß es schon vorbei war. Ich hätte eher erwartet, daß er bei den einzelnen Gebieten einen Streifzug durch alle Kapitel macht. Das hat er aber bei keinem Gebiet gemacht. Er hat sich halt ein oder mehrere Kapitel herausgesucht und dann dazu Fragen gestellt.

Die Reihenfolge der Gebiete hat er übrigens selbst ausgesucht. Ich hatte die Gebiete bei unserer Absprache in einer anderen Reihenfolge angegeben und konnte auch nicht wählen. Ich glaube, da braucht man sich keinen Illusionen hinzugeben, daß man da etwas bewirken könnte.

Bei den Gebieten, die nicht zu seinen eigenen gehören, hier also Termersetzung und Model-Checking fragt er meines Erachtens nur Grundlagen ab. In Model-Checking haben wir z.B. auch sehr lange über symbolisches und TCTL-Model-Checking gesprochen. Ich hatte aber den Eindruck, daß er darauf nie eingegangen wäre. In Termersetzung war es ähnlich. Da meinte er auch nach der Prüfung zu mir: "Hätte ich Sie da auch zu kritischen Paaren und Knuth-Bendix-Ordnungen befragen können?" Als ich meinte, "ja klar, wenngleich Herr Baader Knuth-Bendix-Ordnungen ja praktisch nur erwähnt hat", erwiderte er: "Na ja, da müßte man aber schon weitreichende Kenntnisse haben und deutete an, daß er das wohl nie abfragen würde." Tja, hätte ich vorher wissen müssen! Aber es ist immer besser, mehr zu wissen als zu wenig. Und eins noch: in einem Protokoll habe ich gelesen, daß man gar nicht so viel kotzen könne, wie man gegessen hat. Kann ich nur bestätigen.

In diesem Sinne.

Und viel Glück bei euren Prüfungen!!

Mündliche Diplomprüfung in Theoretischer Informatik

Prüfer: Prof. Oberschelp

Themen: 1. Schaltkreistheorie (Vorl. von Prof. Oberschelp WS 93/94)
2. Komplexitätstheorie (Vorl. von PD Staiger SS94)
3. Termersetzungssysteme (Vorl. von Prof. Baader WS 94/94)

Datum: Dezember 1995

Dauer: 55 min

Note: 1.0

1. Termersetzungssysteme

- Was sind Wortersetzungssysteme?
- Was ist das Wortproblem?
- Welche Probleme sind (im Kontext von Wortersetzungssystemen) unentscheidbar?
- Wie zeigt man, daß das Terminierungsproblem für Wortersetzungssysteme unentscheidbar ist?
- Wie ist die allgemeine Vorgehensweise beim Nachweis der Unentscheidbarkeit eines Problems?
- Warum ist die Unifikation ein entscheidbares Problem?
- Was ist eine Reduktionsordnung?
- Was sind Knuth-Bendix-Ordnungen? Wozu setzt man sie ein?
- Haben Sie den λ -Kalkül in der Vorlesung behandelt?
- Welche Einsatzgebiete für die Termersetzung wurden genannt?

2. Komplexitätstheorie

- Was ist das Regime?
- Da war doch ein Satz darüber, daß unter bestimmten Voraussetzungen an das Regime die von einer Turingmaschine entschiedene Sprache regulär ist. Welche Voraussetzung wird benötigt und wie ist die Beweisidee?
- (Nachfrage zur vorherigen:) Warum haben alle Spuren ungerade Länge?
- Welche Beschleunigungs- und Kompressionssätze kennen Sie?
- Idee der Hierarchiesätze
- Was heißt NP-schwer?
- Wie zeigt man, daß ein Problem in NP liegt?
guess & check