

# Diplomprüfung in theoretischer Informatik

RWTH – Aachen  
Gedächtnisprotokoll

**Fächer:** Compilerbau  
Logik Programmierung  
Angewandte Automaten-Theorie

**Prüfer:** Prof. Indermark  
**Datum:** 06. Juni 2005  
**Dauer:** 50 – 55 min  
**Note:** 1.3

**Sonstiges:** Die Atmosphäre war sehr entspannt, Prof. Indermark sehr fair und der Beisitzer hat nur mitgeschrieben. Da der Lehrstuhl gerade renoviert wurde, waren wir in einem anderen Raum, in dem es leider keine Uhr gab, aber die Zeit verging auch so prima.

## Compilerbau (ca. 20 min):

- Lexikalische Analyse:
  - DTA- und NFA-Methode beschreiben, Komplexität der Konstruktion und der Verarbeitung einer Eingabe für beide Methoden herleiten
  - Backtrack-DFA erläutern, welche Zustände können eliminiert werden?  
(*Nicht erreichbare Zustände + unproduktive zu einem zusammenfassen.*)  
Ist dort noch eine Minimierung möglich?  
(*Wenn die einzelnen DFAs minimiert sind ist der resultierende Backtrack-DFA auch minimiert.*)
- Syntaktische Analyse:
  - SLR(1): Entwerfen Sie eine Analyse um  $G \in \text{SLR}(1)$  festzustellen! Welche Konflikte gibt es? Detaillierte Beschreibung der Konflikte und wie man diese beheben kann.
  - LR(0): Wie kann man  $G \in \text{LR}(0)$  entscheiden?
- Semantische Analyse:
  - LAG: Was sind die Eigenschaften einer LAG? Welche Schritte gibt es im zugehörigen Automaten? Was geschieht beim match-Schritt auf dem Keller?  
(*Die oberste LR(0)-Information wird durch eine andere ersetzt.*)  
Wo ist der Unterschied zum Vergleichsschritt beim normalen TDA?  
(*Hatte keine Ahnung und habe sie noch nicht, war nicht so schlimm.*)  
Detaillierte Erläuterung des expand-Schrittes inklusive dem Index-Shift.
- Zwischencodeerzeugung:
  - „Wie geht denn  $\text{LOAD}(2, 2)$ ?“ Was bedeuten die Argumente? Worauf greift  $\text{LOAD}(2, 2)$  zu? „Erläutern Sie mal anschaulich (ohne die base Funktion) die Modifikation des ZR beim  $\text{LOAD}(2, 2)$ .“

### Logik Programmierung (ca. 15 – 20 min):

- Fixpunktsemantik:
  - Ausführliche Erläuterung von trans! Welche drei Eigenschaften hat trans?  
“Erläutern Sie die Arbeitsweise von trans anhand eines Beispiels, nehmen wir mal die Addition von zwei natürlichen Zahlen.“
    - Logikprogramm für Addition zweier Zahlen definieren.
    - Erstes Ergebnis/Ausgabe für die Anfrage  $\text{add}(X, Y, S0)$  herleiten.
    - Zweites Ergebnis nach einer Iteration von trans für die gleiche Anfrage.
- Herbrand-Strukturen:
  - Was ist der Vorteil bei der Einschränkung auf Herbrand-Strukturen?  
(*Das wusste ich nicht so direkt und an den weiteren Verlauf erinnere ich mich nicht mehr genau, es ging sich aber viel um Herbrand-Strukturen, man kann es sicherlich mit „darauf rumreiten“ umschreiben ...*)
- Anwendung:
  - Angabe des naiven LP zur Listeninvertierung, Bestimmung der Komplexität  
(*Ich wusste, dass es aus  $O(n^2)$  ist und auch, dass es an der verschachtelten Rekursion liegt, aber wo die nun genau steckt konnte ich nicht sagen. Ich gab offen zu, dass ich das beim Lernen nicht gecheckt habe, Prof. Indermark bezeichnete diese Aussage nach der Prüfung als „Eigentor“.  
Es ist aus  $O(n^2)$ , da  $n$ -mal  $\text{append}$  aufgerufen wird mit einer Komplexität  $O(n)$ .)*)
  - Angabe des LP zur Listeninvertierung mit Akkumulator.
- DCG/CFG:
  - Wie ist eine CFG als Logikprogramm darstellbar, also: jedes Nichtterminalsymbol wird einem Prädikat zugeordnet, was ist das für ein Prädikat?  
(*Ein zweistelliges Prädikat.*)  
Übersetzung der Ableitungsregel  $A \rightarrow BC$  in ein Logik Programm.  
“Erklären sie mal ohne die Logik Programmierung, wann eine Tupel  $(w, v)$  akzeptiert wird, wobei  $w$  und  $v$  aus  $\Sigma^*$ .“  
(*Wenn  $v$  ein Suffix von  $w$  ist und das Präfix  $w - v$  aus  $A$  ableitbar ist.*)

### Angewandte Automaten-Theorie (ca. 15 min):

- PDA, PDS: Was ist der Unterschied? Was wird bei PDS ignoriert/vernachlässigt?
- PDS:
  - Wie sehen die Transitionen aus? Ist Erreichbarkeit entscheidbar? Wie ist Erreichbarkeit definiert? Ausführliche Erläuterung der forward-reachability.  
(*Also:  $\text{forward} \rightarrow \text{post}^*(C) \rightarrow 2$ . Saturierungsalgorithmus*)
- PDA:
  - (*Jetzt gings total daneben ...*)  
Jeweils: Entscheidbar? Warum? Wie? Komplexität? Zu
    - Wortproblem
    - Leerheit
    - Äquivalenz von PDA

**Viel Erfolg!!!**