

Prüfungsprotokoll zur Theoretischen Informatik

Christoph Mies

September 27, 2004

Fächer: Compilerbau, Angewandte Automatentheorie, Effiziente Algorithmen
Prüfer: Prof. Indermark
Dauer: ca. 50 Minuten
Note: 1.3

Compilerbau - Basis: Vorlesung

Dauer: ca. 15 Minuten

- Was ist die Aufgabe der Lexikalischen Analyse?
Zerlegung des Eingabestrings in die Lexikalische Struktur, also eine Folge von Symbolen.
- Wie kann man die lexikalische Struktur beschreiben?
Man beschreibt die Struktur mit Regulären Ausdrücken. Diese können dann mit dem Thompson- Alg. in NFA's umgewandelt werden...
- Da habe ich mal ein. Wenn man die Regulären Ausdrücke um Komplement erweitert, wie muss dann der Thompson- Alg. modifiziert werden?
Man muss den NFA mit Potenzmengenkonstruktion determinisieren, dann die akzeptierenden und nicht- akzeptierenden Zustände vertauschen. Dann die neuen akzeptieren Zustände in einen neuen Endzustand mit ϵ - Transitionen überführt werden. (Da gab es auch eine Übungsaufgabe zu).
- Ok. Wie geht Scannerkonstruktion weiter?
Konstruktion des Vereinigungsautomaten. Modifizieren der Endzustandstapel, um first- match zu erhalten. Dann Backtrack- Automat konstruieren.
- Soviel zur lexikalischen Analyse. Jetzt zur syntaktischen Analyse. Wie sind LR(1)- Auskünfte definiert?
Definition angeben.
- Zur Faltung von LR(1) zu LALR(1). Welche Konflikte können auftreten und warum?
Nur reduce-reduce Konflikte. Grund: LR(0)- Kern ist gleich, daher kann kein shift-reduce Konflikt auftreten, weil er sonst schon in LR(1) aufgetreten wäre.
- Gut. Kann ich Sie auch zur Übersetzung von Prozeduren mit Parametern befragen?
Klar.
- Was geschieht bei der Anlegung eines neuen Frames?
Zuerst kurz neue Stack- Maschine erläutert: Ein Stack, IC- Register, Stack-, Frame- und Index- Pointer... Wird mit dem Call-Befehl gestartet. Dann werden die Parameter ausgerechnet, der statische Verweis wird mit Hilfe des Indexregisters berechnet und eingetragen... Das hab ich zumindest gesagt, schien auch nicht vollkommen falsch zu sein...

Angewandte Automatentheorie - Basis: Skript von 2003

Dauer: ca. 25 Minuten

- Zuerst die PDSs. Wie sehen die aus?
Definition angegeben, insbesondere Transitionsrelation.
- Dann zu den Entscheidbarkeitsproblemen. Es gibt den Reachability Tree, der i.A. unendlich ist. Ist Reachability Problem Problem entscheidbar?
Ja, das ist entscheidbar. Habe was zu forward und backwards reachability gesagt und den ersten Saturierungsalgorithmus erläutert.
- Ok. Nun die Unentscheidbaren Dinge. Warum ist die Erreichbarkeit bei einem 2-PDS unentscheidbar?
Man kann eine Turingmaschine auf dieses Problem reduzieren. Kurz die Beweisidee beschrieben.
- Was ist ein Counter System?
Ein PDS mit nur einem von Z_0 verschiedenen Kellersymbol.
- Ja. Ist mit der Einschränkung das Erreichbarkeitsproblem entscheidbar?
Nein. 2-PDS kann auf 4-CS reduziert werden und 2-PDS auf 1-CS. Ich brauchte aber nichts genaueres dazu zu sagen.
- Kommen wir zu Baumautomaten. Was ist der Unterschied zwischen XML-Automaten und normalen Baumautomaten?
Beliebiger Branching Faktor. Sonst gelten alle Ergebnisse der konventionellen Baumautomaten.
- Ok. Wie sieht ein Baumautomat aus?
Habe die Definition angegeben. Konnte aber die Definition von δ^* nicht richtig angeben. Prof. Indermark musste mir extrem viel helfen, bis ich es endlich hinbekommen habe. Habe unter anderem dabei auch noch Baumautomat für die Sprache angegeben, die im Frontier erst nur a's und dann nur b's stehen hat (Steht auch im Skript). Hab mit hier wohl die 1.0 verbaut.

Effiziente Algorithmen - Basis: Buch

Dauer: ca. 10 Minuten

- Für manche Probleme ist es ja echt schwierig, effiziente Algorithmen zu finden. Welche meine ich wohl?
Wahrscheinlich die NP- Vollständigen.
- Richtig. Nennen Sie mal welche!
SAT, TSP ...
- Was ist denn SAT?
Erfüllbarkeitsproblem für Boolesche Formeln.
- Wie sieht ein naiver Algorithmus aus?
Einfach alle Belegungen ausprobieren. Worst Case Aufwand: $O(2^n)$.
- Warum muss die Logikprogrammierung nicht mit diesem Aufwand arbeiten (auch wenn es nicht zu Ihrer Prüfung gehört)?
Bei LP werden nur Horn- Formeln betrachtet.
- Was ist VCP?
Erklärt.
- Wieviele Möglichkeiten gibts es denn, bei gegebenem k , ein Vertex Cover der Grösse k zu finden?
Man muss alle Teilmengen betrachten. Also $\binom{n}{k}$ viele.

- Ist das jetzt polynomial oder exponentiell?
Wusste ich nicht, hab dann die Formel aufgelöst und bin dann auf $O(n^k)$ gekommen (mit Hilfe vom Prof.). Für jede Teilmenge muss dann der Test mit Komplexität von $O(|E|)$ durchgeführt werden. Dann hat das also polynomialen Aufwand.
- Zum Schluss noch: Welchen Aufwand haben gute Suchalgorithmen?
 $O(n \cdot \log(n))$. Z.B. Quicksort.
- Gibt es noch welche?
Mergesort.
- Ok, warten Sie bitte kurz draussen!

Fazit

Insgesamt war es eine angenehme Atmosphäre. Ich war vorher sehr nervös. Doch als ich anfang zu reden, legte sich die Nervösität schnell. Der Beisitzer, Michael Weber, sagte kein Wort. Habe mich also nur mit Prof. Indermark unterhalten.

Als die Prüfung zuende war und ich draussen wartete, dachte ich eigentlich, ich hätte eine schlechtere Note. Der Compilerbauteil war wohl sehr gut. Der Automatenteil war bis zu letzten Frage auch sehr gut. Im letzten Abschnitt wurde alles sehr schnell abgehakt. Ich weiss nicht, wie gut ich da war. Insgesamt ist die Note doch sehr fair.