

Generell sehr angenehme Atmosphäre.

Herr Schulz war ein sehr angenehmer Prüfer, am Ende haben wir noch ein bisschen über den Kurs selbst geredet und noch eine kurze Diskussion über die Effizienz der Algorithmen zur Bestimmung von FOLLOW/FIRST geführt, da ich es anders erklärt hatte als im Skript. Es wird darauf Wert gelegt, dass man die zusätzlich zu den mündlichen Antworten auch viel auf dem Whiteboard zeichnet. Wir haben relativ lange bei den Top-Down Sachen gebraucht, weil ich einen anderen Algorithmus erklärt habe, deswegen war am Ende für manche Sachen denke ich keine Zeit mehr, speziell zu 3AC wäre sonst glaube ich mehr gefragt worden.

1. „Die Übersetzung ist unterteilt in Analyse / Synthese. Nennen Sie doch mal die 3 Phasen der Analyse“?

Lexikalische Analyse, syntaktische Analyse, semantische Analyse

2. „Was macht man bei der lexikalischen Analyse genau“?

Input String wird zu Tokens umgewandelt. Token mit Beispiele (Bezeichner, Keywords, Konstanten,...)

3. „Es geht also darum Tokens zu erkennen – zeichnen und erklären Sie einen Automat der Integer erkennt.“

Automat (ist kein DEA übrigens) gezeichnet der Integer erkennt. Ich hab einen gezeichnet der noch 0 unterscheidet (keine führende 0) und halt erklärt.

4. „Was macht man bei der syntaktischen Analyse genau“?

Syntax wird analysiert, also wie die Token zusammenhängen. Input: Tokenmenge Output: Syntaxbaum

5. „Welche Verfahren gibt es zur Syntaxanalyse“?

Top-Down und Bottom-Up

6. „Was zeichnet das Top Down Verfahren aus“?

Syntaxbaum wird von der Wurzel aus mit dem Startsymbol der Grammatik entwickelt.

7. „Wie funktioniert das konkret“?

Als Beispiel Recursive Descent Verfahren erklärt.

8. „Welche Probleme können hier auftreten“?

Linksrekursion und dadurch Endlosschleife, Beispielproduktionsregel aufgeschrieben

9. „Wie kann man das Problem der Linksrekursion lösen“?

Wie im Skript Linksrekursion auflösen am Beispiel schriftlich gezeigt.

10. „Linksfaktorisierung – welches Problem löst das und zeigen Sie eine“

Gleiche Terminalsymbole bei verschiedenen Optionen -> keine Entscheidung durch Lookahead möglich.

Wieder schriftliches Beispiel gezeigt und aufgelöst.

11. „Was heißt eigentlich LL(k) und was ist das für eine Grammatik“?

Formale Definition (Steuermengen paarweise disjunkt) + Erklärung der beiden Ls (von links, Linksableitung) und k(Lookahead)

12. „Was ist die schon angesprochene Steuermenge“?

Steuermenge ist Konkatenation von FIRST und FOLLOW, FIRST und FOLLOW an Beispielregeln erklärt ohne formale Definition

13. „Wie berechnet man diese Mengen“?

Algorithmen zur Berechnung der Mengen erklären – ich hatte hier bei FOLLOW einen anderen Algorithmus als aus dem Skript erklärt, der wohl ineffizienter war, aber auch korrekt. Diese Frage ging relativ lang, sollte man also auf jeden Fall erklären können!

14. „Bottom-Up Verfahren – was zeichnet dieses aus“?

Baum wird von den Blättern (Tokens) aus aufgebaut.

15. „Bei Bottom Up benutzt man Shift-Reduce Parser, welche Aktionen gibt es und wie funktioniert das“?

Die vier Aktionen Shift, Reduce, Accept, Error erklärt – kleines Beispiel mit Shift und Stack aufgezeichnet.

16. „Was versteht man unter einem Handle“?

Formale Definition aufgeschrieben und erklärt.

17. „Erklären Sie kurz die Operatorvorranganalyse.“

Shift / Reduce Konflikt erklärt -> macht Entscheidung notwendig. Die drei Relationen kurz erklärt, Aufstellen der Tabelle, vor jeder Aktion wird die Relation auf dem Stack mit zu lesendem Token verglichen und jeweilige Aktion durchgeführt.

18. „Was sind die Anforderungen dafür, dass die Operatorvorranganalyse möglich ist“?

Es gibt 3 Kriterien (ich wusste nur eps): 1. Keine Epsilon Produktion, 2. Keine aufeinanderfolgenden Terminale rechte Seite, 3. Keine gleichen rechten Seiten.

19. „Welche 3 Phasen hat die Synthese“?

Zwischencodgenerierung, Codeoptimierung, Codegenerierung

20. „Was kann man sich unter Zwischencode vorstellen“?

Beispiele: AST, DAG, Postfix, 3AC

21. „3 Adress Code – können Sie kurz erklären was das ist?“

An Assembler angelegter Code bei dem maximal 3 Adressen auftreten können.

22. „Nennen Sie je ein Beispiel mit Erklärung für maschinenabhängige / -unabhängige Optimierung.“

Constant Folding / Propagation als unabhängige Optimierung mit Beispiel erklärt.

Peephole Optimisation mit Beispielen aus Skript (LOAD Zeile im Assembler sparen, überflüssige GOTO Anweisung) erklärt und aufgeschrieben

23. „Wieso heißt das eigentlich Peephole Optimization?“

Weil nur ein sehr kleiner zusammenhängender Ausschnitt des Maschinencodes betrachtet wird.

Bitte schreibt auch Prüfungsprotokolle für eure mündlichen Prüfungen - das hilft anderen Studierenden wahnsinnig für die Vorbereitung und ist nicht viel Arbeit!