

Hinweise zur Bearbeitung der Klausur zum Kurs 1810 „Übersetzerbau“

Bitte lesen Sie sich diese Hinweise vollständig und aufmerksam durch, bevor Sie mit der Bearbeitung der Klausur beginnen.

1. Die Klausurdauer beträgt 3 Stunden.
2. Prüfen Sie bitte die Vollständigkeit Ihrer Unterlagen. Die Klausur umfaßt
 - 2 Deckblätter
 - diese Hinweise
 - 1 Formblatt für eine Teilnahmebescheinigung zur Vorlage beim Finanzamt
 - 6 Aufgaben auf den Seiten 2 - 5
3. Bevor Sie mit der Bearbeitung der Klausuraufgaben beginnen, füllen Sie bitte die folgenden Teile der Klausur aus:
 - (a) *sämtliche* Deckblätter mit Name, Anschrift sowie Matrikelnummer. Markieren Sie vor der Abgabe auf allen Deckblättern die von Ihnen bearbeiteten Aufgaben.
 - (b) die Teilnahmebescheinigung, falls Sie diese wünschen.
4. Schreiben Sie Ihre Lösungen auf Ihr eigenes Papier (DIN A4) und nicht auf die Seiten mit den Aufgabenstellungen. Heften Sie vor Abgabe der Klausur die Deckblätter, die Klausuraufgaben und ggf. die Teilnahmebescheinigung an Ihre Bearbeitung.
5. Schreiben Sie bitte auf jedes Blatt oben links Ihren Namen und oben rechts Ihre Matrikelnummer. Numerieren Sie Ihre Seiten bitte durch.
6. Als Hilfsmittel sind nur unbeschriebenes Konzeptpapier und Schreibzeug (kein Bleistift!) zugelassen.

Es sind maximal 100 Punkte erreichbar. Wenn Sie mindestens 50 Punkte erreicht haben, können Sie davon ausgehen, daß Sie die Klausur bestanden haben.

Aufgabe 1 (Endliche Automaten)

20 Punkte

Geben Sie für jede der folgenden Sprachen einen endlichen, deterministischen Automaten an, der diese Sprache erkennt. Beachten Sie, daß in Zahlendarstellungen keine führenden Nullen erlaubt sind. Sie können die Zustandsübergänge wahlweise als Diagramm oder als Tabelle angeben.

- (a) $L_1 = \{ w \in \{0-9\}^* \mid w \text{ ist eine gerade Dezimalzahl} \}$ 4 Punkte
- (b) $L_2 = \{ w \in \{0-9\}^* \mid w \text{ enthält genau eine 1 und höchstens zweimal die 8} \}$ 7 Punkte
- (c) $L_3 = \{ w \in \{0-9\}^* \mid w \text{ ist eine durch 7 teilbare Dezimalzahl} \}$ 9 Punkte

Aufgabe 2 (LL(1)-Grammatik)

20 Punkte

Gegeben sei die folgende, kontextfreie Grammatik
 $G = (\{U, V, W, X, Y, Z\}, \{a, b, c, d, e, f, g, h\}, P, U)$,

$$\text{mit } P = \left\{ \begin{array}{l} U \rightarrow V \mid UaV \\ V \rightarrow W \mid bUb \\ W \rightarrow X \mid WcX \\ X \rightarrow d \mid eWe \\ Y \rightarrow fW \mid fbUb \mid ghV \\ Z \rightarrow ghV \mid Y \end{array} \right\}.$$

- (a) Geben Sie eine äquivalente Grammatik G' mit LL(1)-fähigen Produktionen an. 6 Punkte
- (b) Berechnen Sie die FIRST-Mengen 4 Punkte
- (c) Begründen Sie anhand der Steuermengen, ob G' vom Typ LL(1) ist. Bei der Berechnung der FOLLOW-Mengen führen Sie bitte alle relevanten Produktionen in einer Tabelle auf und geben Sie zu jedem Eintrag evtl. resultierende Knotenmarkierungen und Kantenbeziehungen an. Zeichnen Sie danach den Graphen und kennzeichnen Sie dort ggf. starke Komponenten. Das Resultat der Propagation von Knotenmarkierungen soll nicht eingezeichnet werden, geben Sie stattdessen direkt die FOLLOW-Mengen an. 10 Punkte

Aufgabe 3 (Operator-Vorrang-Analyse)**16 Punkte**

- (a) Erstellen Sie für folgende Operatoren die Tabelle mit den Vorrangrelationen.
- 8 Punkte**

 $\otimes, \odot, \otimes, \oplus$

Dabei hat \otimes eine höhere Priorität als \odot und höhere Priorität als \otimes usw. \oplus ist im Gegensatz zu den anderen Operatoren rechtsassoziativ.

Neben den angegebenen Operatoren gibt es natürlich noch $(,), \text{id}$ und S , die sich wie gewohnt verhalten.

- (b) Parsen Sie mit der Tabelle aus (a) den Ausdruck
- 8 Punkte**

 $(\text{id} \oplus \text{id} \oplus \text{id}) \odot \text{id} \otimes \text{id}$

unter Verwendung folgender Grammatik:

$$G = \{ \{E\}, \{ \text{id}, \otimes, \odot, \otimes, \oplus, (,) \},$$

$$\{ E \rightarrow (E),$$

$$E \rightarrow E \otimes E,$$

$$E \rightarrow E \oplus E,$$

$$E \rightarrow E \odot E,$$

$$E \rightarrow E \otimes E,$$

$$E \rightarrow \text{id} \}$$
Aufgabe 4 (Römische Zahlen)**16 Punkte**

Die römischen Zahlen werden durch die folgenden (stark vereinfachten) Regeln gebildet.

- Jedes Zeichen besitzt den in der nachfolgenden Tabelle angegebenen Zahlenwert.
- Zeichen werden mit Ausnahme der nächsten Regel additiv verknüpft.
- Folgt einem Zeichen ein anderes Zeichen mit einem größeren Wert, so ist der Wert des ersten Zeichens vom Wert des zweiten Zeichens abzuziehen, um den Gesamtwert beider Zeichen zu erhalten.

Zeichen	M	D	C	L	X	V	I
Wert	1000	500	100	50	10	5	1

- (a) Geben Sie eine attributierte Grammatik G an, mit deren Hilfe sich der Wert einer römischen Zahl bestimmen läßt. Gehen Sie davon aus, daß die Symbole M, \dots, I die Terminale der Grammatik bilden. 10 Punkte
- (b) Geben Sie den Datenflußgraphen für die Zahl $MCMLXIV$ an. 6 Punkte

Aufgabe 5 (3-Adreß Code)

12 Punkte

Gegeben sei folgendes Programm der im Kurstext eingeführten, imperativen Beispielsprache:

```
program main {  
  
  procedure compute(a : integer; b : integer) : integer {  
    var integer x;  
    x := a + b * b;  
    return x;  
  }  
  
  var integer x;  
  const integer a := 5;  
  const integer b := 7;  
  
  x := compute(a, b);  
  ...  
}
```

- (a) Übersetzen Sie das Hauptprogramm und die Prozedur in 3-Adreß Code. 4 Punkte
- (b) Für den Aufbau eines Aufruf-Stack sind die Inhalte von Symboltabelle und Prozedurtabelle von Bedeutung. Welche Inhalte haben diese Tabellen für das Programm aus Teil (a)? 8 Punkte
- Erläutern Sie die darin benötigten Informationen und gehen Sie davon aus, daß der Index der Typtabelle für den Typ Integer den Wert 1 hat, und die Größe der Darstellung sowie das Alignment von Variablen dieses Typs jeweils den Wert 4 besitzen. Die Startadresse der Funktion `compute` sei 1456.

Aufgabe 6 (Maschinen-Code-Optimierung)**16 Punkte**

Gegeben sei folgendes Programmstück in Maschinen-Code:

```
1      LOAD  S,10
2      STORE S,y
3      LOAD  S,2
4      STORE S,t
5      LOAD  T,0
6      STORE T,z
7      STORE T,x
8 L1:  if x > y goto L2
9      LOAD  T,x
10     ADD   T,1
11     STORE T,x
12     LOAD  T,z
13     LOAD  U,t
14     MULT  T,U
15     STORE T,z
16     goto  L3
17 L2:  LOAD  T,z
18     LOAD  U,z
19     MULT  T,U
20     goto  L4
21 L3:  goto  L1
22 L4:  STORE T,z
```

Der Maschinen-Code sei entstanden aus einer Übersetzung von 3-Adreß-Code für einen bestimmten Prozessor-Typ. Nun steht ein Prozessor $Proc^{EXP}$ zur Verfügung, der in seinem Befehlssatz zusätzlich die Operationen $SHLFT\ R$ für binäres Links-Shift auf einem Register und $INC\ x$ für die Inkrementierung eines Integer-Wertes auf einem beliebigen Speicherort besitzt.

- (a) Wie sieht das Programm aus, wenn direkt für $Proc^{EXP}$ übersetzt worden wäre. **4 Punkte**
- (b) Optimieren Sie das Ergebnisprogramm aus (a) mittels wiederholter Peephole-Optimierung mit einem Ausschnitt der Größe 8. **10 Punkte**
- (c) Welcher Wert steht am Ende des Programms in z ? **2 Punkte**