

Name		Kennzahl	Matrikelnummer
		Bitte tragen Sie sofort mit Kugelschreiber <i>Name, Kennzahl</i> und <i>Matrikelnummer</i> ein! Halten Sie bitte den Studentenausweis bereit!	
Summe		Lesen Sie die Angaben genau durch, bevor Sie beginnen! Verwenden Sie bitte nur diesen Bogen für Ihre Antworten! Die Prüfungsdauer beträgt 90 Minuten.	
1.Bsp.		Institut für Computersprachen	
2.Bsp.			
3.Bsp.			
4.Bsp.			

Ich trete zum ... mal zur Prüfung an.

Prüfung aus Übersetzerbau 24.6.2004

1. 20 % Grammatik

Gegeben sei folgende Grammatik (Zeichenketten in Hochkomma sind Terminalsymbole):

$$\begin{aligned}
 S &\rightarrow \text{'[' A L ']'} \\
 L &\rightarrow \text{' , ' A L} \\
 L &\rightarrow \varepsilon \\
 A &\rightarrow B \\
 A &\rightarrow \text{'[' B ' | ' B ']'} \\
 B &\rightarrow \text{'a'} \\
 B &\rightarrow \text{'[' A L ']'}
 \end{aligned}$$

a) (55 %) Bestimmen Sie die first- und follow-Mengen aller Nonterminale.

b) (5 %) Begründen Sie, warum die Grammatik nicht LL(1) ist.

2. 25 % Gegeben sei folgendes Modula-2-Programm:

```
MODULE m;
TYPE IntProc=PROCEDURE(VAR INTEGER);
TYPE IntProcProc=PROCEDURE(IntProc, VAR INTEGER);
VAR a, b:INTEGER;

    PROCEDURE x(b:IntProc; VAR y:INTEGER);
    BEGIN
        b(y);
    END x;

    PROCEDURE y(VAR c:INTEGER);
    BEGIN
        b:=a+1; a:=c; c:=b;      (*here*)
    END y;

    PROCEDURE z(c:IntProcProc; d:INTEGER);

        PROCEDURE p(d:IntProc; VAR e:INTEGER);
        BEGIN
            c(d,e);
        END p;

    BEGIN
        p(y,d);
    END z;
BEGIN
    a:=1; b:=3;
    z(x,a);
END m.
```

Zeichnen Sie den Stack der Activation Records (mit lokalen Größen, Wert- und Variablenparametern, statischer und dynamischer Kette) zum Zeitpunkt (*here*) ohne Verwendung eines Displays.

3. 25 % Befehlsanordnung

Gegeben sei folgendes Maschinenprogrammstück in Alpha-ähnlicher Syntax (`op rs1,rs2,rd` oder `ldl rd,offset(rs1)` oder `stl rs1,offset(rs2)`). Ein `ldl`-Befehl dauert 2 Zyklen, ein `mull`-Befehl dauert 3 Zyklen, alle anderen Befehle dauern einen Zyklus. Geben Sie den Datenabhängigkeitsgraph an, tragen Sie die Kantenlängen ein, bestimmen sie die Pfadlängen im Datenabhängigkeitsgraphen und geben Sie die optimierte Befehlsanordnung an, die Sie mittels *list scheduling* erhalten.

```
A  addl  s0,s1,t1
B  mull  t1,s2,t1
C  addl  s0,s2,t2
D  mull  s3,4,t3
E  addl  t3,s4,t3
F  ldl   t4,0(s5)
G  stl   t2,0(t3)
H  ldl   t5,0(s0)
I  addl  t1,t4,t1
K  addl  t1,t5,t1
L  stl   t1,0(s0)
```

4. 30 % Attributierte Grammatik

Ein Autokatalog mit Nonterminal **A** besteht aus einer Liste von Autos. Es gibt mindestens ein Auto. Nach der Autobezeichnung "**auto**" stehen die Extras. Jedes Auto hat mindestens ein Extrateil. Ein Extrateil **E** hat einen Extranamen "**extra**" und die Anzahl "**num**" der Extras, die für dieses Auto benötigt werden. Ein Auto ist durch "." abgeschlossen, z.B.:

```
Passat Diesel 1 Tuere 4 Radio 1.  
Golf Benzin 1 Tuere 5 Lederlenkrad 1.
```

a) (8 %) Erstellen Sie eine Grammatik, die einen ganzen Katalog beschreibt. (Verwenden Sie dazu die Terminalsymbole "**auto**", "**extra**", "**num**" und "." und ignorieren Sie etwaige Leerzeichen.)

b) (17 %) Erweitern Sie die Grammatik aus a) um Attribute zur Berechnung des Preises eines Autos. Die Ausgabe soll so wie das Eingabeformat sein, nur dass zusätzlich nach der Autobezeichnung der Preis in Euro steht. Die Ausgabe soll im synthetisierten Attribut "**A.preis**" erfolgen. Im Attribut "**extra.p**" steht der Preis für einen Extra in Euro, in "**auto.n**" und "**extra.n**" steht der Name, und in "**num.z**" der Zahlenwert des entsprechenden Terminalsymbols. Der Operator "|" hängt Zeichen- und Zahlen intelligent aneinander ("**Diesel**"|1 ergibt "**Diesel 1**"). Das obige Beispiel ergibt (**Diesel.p=20000; Benzin.p=15000; Tuere.p=800; Radio.p=300; Lederlenkrad.p=200;**):

```
Passat 23500 Diesel 1 Tuere 4 Radio 1.  
Golf 19200 Benzin 1 Tuere 5 Lederlenkrad 1.
```