

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

Ich trete zum ... mal zur Prüfung an.

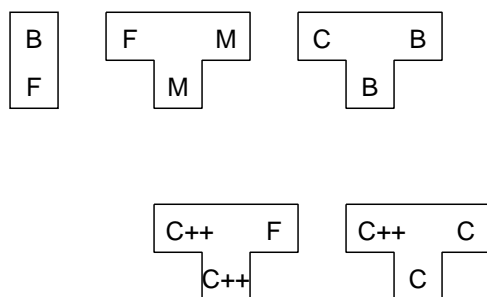
## Prüfung aus Übersetzerbau 10.11.1995

### 1. 20 % T-Diagramme

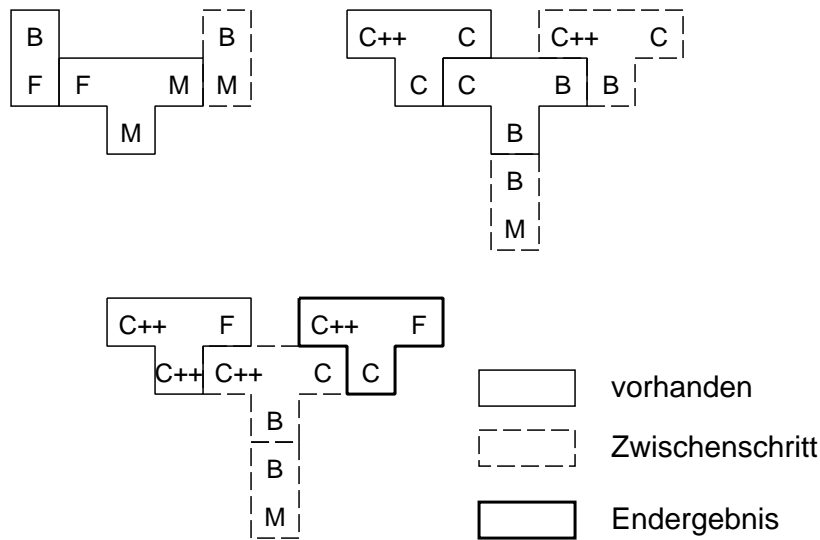
Sie verfügen über einen in Fortran geschriebenen Basic-Interpreter, einen Compiler von Fortran für die Maschine M, der Maschinencode erzeugt, einen in Basic geschriebenen Compiler, der C in Basic übersetzt, einen in C++ geschriebenen Compiler, der C++ in Fortran übersetzt und einen in C geschriebenen Compiler, der C++ in C übersetzt.

Sie haben nur eine Maschine M zur Verfügung.

a) (5 %) Stellen Sie die vorhandenen Compiler und Interpreter mittels T- und I-Diagrammen dar.



b) (15 %) Erzeugen Sie (mittels T- und I-Diagrammen) einen in C geschriebenen C++-Compiler, der Fortran erzeugt.



## 2. 25 % Reguläre Ausdrücke

a) (15 %) Erstellen Sie eine reguläre Definition für Pfadnamen. Ein Pfadname besteht aus Filenamen, die (wie in UNIX) durch den Pfadseparator “/” getrennt sind und endet mit einem Filenamen. Es dürfen nie zwei Pfadseparatoren direkt hintereinander stehen. Ein Filename kann aus Klein- und Großbuchstaben, aus dem Sonderzeichen “.” und aus Ziffern bestehen.

Beispiele für gültige Filenamen:

```
paper.doc
/usr/bin/csh
../../../../Help
./1234
```

Beispiele für ungültige Filenamen:

```
../paper.doc
/usr/bin/#frog
/usr/tmp/
```

```
file = ([a-z] | [A-Z] | [0-9] | ".")+
pfad = /?(file/)*file
```

b) (10 %) Gegeben sind der folgende Text:

EINE FEE, DIE KAFFEE TRINKT,  
IST EINE KAFFEEFEE,  
EINE FEE, DIE FEENKAFFEE TRINKT,  
IST EINE FEENKAFFEEFEE.  
(A FEE, DIE KAN KAFFEE TRINKT,  
IS A KAKAFFEEFEE.)

die beiden regulären Ausdrücke R1 und R2:

R1 = (F+E\*)+  
R2 = N+E<sub>□</sub>T

und die beiden Ersetzungen S1 und S2:

S1 = NONNE  
S2 = NEN<sub>□</sub>T

Ersetzen Sie zuerst alle Textstellen, die mit R1 in Übereinstimmung gebracht werden können, durch S1 und danach alle Textstellen, die mit R2 in Übereinstimmung gebracht werden können, durch S2. Achtung: Immer Zeichenketten maximaler Länge ersetzen! (Anmerkung zur Notation: “<sub>□</sub>” steht für ein Leerzeichen!)

Schreiben Sie den Ergebnistext in das nachfolgende Kästchen (Durchstreichungen und sonstige Veränderungen im obigen Originaltext gelten *nicht* als richtige Lösung)!!!

EINE NONNE, DIE KANONEN TRINKT,  
IST EINE KANONNE,  
EINE NONNE, DIE NONNENKANONEN TRINKT,  
IST EINE NONNENKANONNE.  
(A NONNE, DIE KAN KANONEN TRINKT,  
IS A KAKANONNE.)

### 3. 25 %    Quadrupel-Code

```
VAR
  a: ARRAY[10,20] OF INTEGER;
  i,j,f: INTEGER;
:
WHILE i <= 9 DO
  IF a[i,j] > 5 AND j < 10 THEN
    f := j;
  ELSE (* do nothing *)
  END
  i := i + 1;
END
```

a) (20 %) Erzeugen Sie für das obige Programmstück Quadrupel-Code nach der Kontrollflußmethode. Ein INTEGER ist 4 Bytes groß.

```
begin:if i<=9 goto Wtrue
      goto Wfalse
Wtrue:t1 := 20 * i (* In der WHILE-Schleife *)
      t2 := t1 + j
      t3 := 4 * t2
      t4 := t3 + adr(a)
      t5 := @t4
      if t5>5 goto I1 (* Erste Bedingung *)
      goto Ifalse
I1:   if j<10 goto Itrue (* Zweite Bedingung *)
      goto Ifalse
Itrue:f := j (* THEN-Zweig *)
      goto Iend
Ifalse(* do nothing *)
Iend: t6 := i + 1
      i := t6
      goto begin
Wfalse:
```

**b) (5 %)** Schreiben Sie eine optimierte Version des vorhin erzeugten Codes (eliminieren Sie unnötige Sprungbefehle).

```
L1:   if i>9 goto L3
      t1 := 20 * i (* In der WHILE-Schleife *)
      t2 := t1 + j
      t3 := 4 * t2
      t4 := t3 + adr(a)
      t5 := @t4
      if t5<=5 goto L2 (* Erste Bedingung *)
      if j>=10 goto L2 (* Zweite Bedingung *)
      f := j (* THEN-Zweig *)
L2:   t6 := i + 1
      i := t6
      goto L1
L3:
```

#### 4. 30 % Codeerzeugung

Erzeugen Sie für den folgenden Zwischencode-Baum optimalen, d.h. schnellsten und kürzesten Maschinencode für den 68000, sodaß das Ergebnis in einem Datenregister landet. Verwenden Sie dazu die im Skriptum angegebene Befehlsauswahl-Baumgrammatik. (Zur Erinnerung: Bei gleichen Kosten können Sie willkürlich eine Alternative auswählen.)

