

## Klausur "Compilerbau" – 6.1.2005

Nachname:  
Matrikelnummer:

Vorname:  
Studienordnung (PO 91/PO 02):

Aufgabe	Punktzahl maximal	Punktzahl erreicht
1	4	
2	4	
3	4	
4	4	
5	4	
6	2	
7	4	
Summe	26	

**Bitte geben Sie die Lösungen möglichst direkt in den Aufgabenblättern an.**

**Bearbeitungszeit: 60 Minuten**

**Zur Bearbeitung sind keine Hilfsmittel erlaubt.**

Aufgabe 1 (4 Punkte)

Punkte  von 4

Sei  $G = (N, T, P, S)$  eine kontextfreie Grammatik. Definieren Sie den Begriff „Ableitbarkeit“ bezüglich  $G$ .

## Aufgabe 2 (4 Punkte)

Punkte  von 4

Transformieren Sie nachfolgende Grammatik  $G$  in eine äquivalente LL(1)-Grammatik.

$G = (N, T, P, S), N = \{S, X, Y\}, T = \{a, b, c\}, P :$

$S \rightarrow abX \mid ab$

$X \rightarrow c \mid Yc$

$Y \rightarrow ab \mid Ybc$

## Aufgabe 3 (4 Punkte)

Punkte  von 4

Bestimmen Sie zu folgender Grammatik die LL(1)-Parsertabelle

$S \rightarrow X \mid Yc$

$X \rightarrow aX \mid bY$

$Y \rightarrow dY \mid \epsilon$

## Aufgabe 4 (4 Punkte)

Punkte  von 4

Betrachten Sie folgende Yacc/Bison-Grammatik

%%

S: 'x' A 'x' | 'x' B 'c';

A: 'a' | ;

B: 'b' 'b' B | 'c';

Der Anfangszustand des generierten LR-Parsers ist wie folgt:

Zustand 0

'x'	schiebe und gehe zu Zustand 1 über
S	gehe zu Zustand 11 über

a) Bestimmen Sie den Zustand 1 mit allen Aktionen

b) Nehmen Sie an, die Eingabe lautet „xx“. Mit dem ersten x gelangt der Parser in Zustand 1. Welche Aktion wird in diesem Zustand ausgeführt? Bestimmen Sie den durch diese Aktion erreichten Folgezustand.

### Aufgabe 5 (4 Punkte)

Punkte  von 4

Bestimmen Sie die abstrakte Syntax zum nachfolgenden SPL-Programm als Graph oder unter Verwendung der Baumkonstruktoren des Referenzcompilers oder gemäß der Ausgabekonvention des Referenzcompilers (`spl --absyn`).

```
proc main(){
  var i: array[3] of int;
  i[1] := 5;
  printi(i[1]);
}
```

### Aufgabe 6 (2 Punkte)

Punkte  von 2

Bestimmen Sie das Frame-Layout (Bestandteile in der richtigen Reihenfolge mit Größe in Bytes) für den Aktivierungsrahmen der nachfolgenden SPL-Prozedur *x*.

```
proc x (ref i:int) {  
  var k: int;  
  k:=i+1;  
  i:=k;  
  printi(i);  
}
```

### Aufgabe 7 (4 Punkte)

Punkte  von 4

Bestimmen Sie den ECO32-Assemblercode zur nachfolgenden SPL-Prozedur *x*. Die Prozedur *printi* erwartet einen Wertparameter vom Typ *int*. (SP=\$29, FP=\$25, RET=\$31, verfügbare Register: \$8-\$15).

```
proc x (ref i:int) {  
  var k: int;  
  k:=i+1;  
  i:=k;  
  printi(i);  
}
```