

---

**Prüfungsteilnehmer**

**Prüfungstermin**

**Einzelprüfungsnummer**

---

**Kennzahl:** \_\_\_\_\_

**FRÜHJAHR**

**66110**

**Kennwort:** \_\_\_\_\_

**1993**

**Arbeitsplatz-Nr.:** \_\_\_\_\_

---

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**

**- Prüfungsaufgaben -**

**Fach:** Informatik (vertieft studiert)

**Einzelprüfung:** Automatentheorie, Algorithm. Sprachen

**Anzahl der gestellten Themen (Aufgaben):** 1

**Anzahl der Druckseiten dieser Vorlage:** 4

**bitte wenden!**

Sämtliche Teilaufgaben sind zu bearbeiten!

### 1. Formale Sprachen

- (1a) Sei  $G = (T, N, P, S)$  eine Chomsky-Grammatik. Was bedeuten die angegebenen Komponenten? Wie sind die Produktionsmenge  $P$  und der Ableitbarkeitsbegriff definiert?
- (1b) Welche formalen Einschränkungen auf  $P$  führen zu den monotonen, kontextsensitiven und kontextfreien Grammatiken? Wie liegen die Klassen der erzeugten Wortmengen zueinander? Soweit sie nicht gleich sind, geben Sie (ohne Beweis) je ein typisches Beispiel aus der Differenzmenge an!
- (1c) Für kontextfreie Grammatiken gilt das Zerlegungslemma. Formulieren und beweisen Sie dieses Lemma!
- (1d) Beweisen Sie, daß sich die Wortmenge

$$\{a^i b^j \mid i, j \geq 0 \wedge i \neq j\}$$

von einer kontextfreien Grammatik erzeugen läßt!

### 2. Berechenbarkeit

- (2a) Der Berechenbarkeitsbegriff kann nach Kleene durch rekursive Funktionen definiert werden. Geben Sie die Definition der primitiv-rekursiven und der  $\mu$ -rekursiven Funktionen an!
- (2b) Zeigen Sie, daß der beschränkte  $\mu$ -Operator nicht aus dem Bereich der primitiv-rekursiven Funktionen hinausführt!
- (2c) Beweisen Sie, daß die ganzzahlige Quadratwurzelfunktion  $\lfloor \sqrt{n} \rfloor$  primitiv-rekursiv ist!
- (2d) Die  $\mu$ -rekursiven Funktionen lassen sich unmittelbar in *while*-Programme übersetzen, die nur aus
- Wertzuweisungen,
  - dem Aufruf der Nachfolger- und Vorgängerfunktion,
  - *while*-Schleifen und
  - Prozeduraufrufen
- bestehen. Zeigen Sie, daß es kein *while*-Programm mit genau einer Variablen gibt, das die Funktion  $f(x) = 2x$  berechnet!

### 3. Algorithmische Sprachen

- (3a) Eine Objektart ist durch eine Trägermenge und einen Satz Operationen definiert. Wie entsteht die Trägermenge bei einer Verbundart und welche Operationen sind darauf definiert? Verwenden Sie als Beispiel:

```
mode datum = (int [1..31] tag,
               int [1..12] monat,
               int [1900..1999] jahr)
```

- (3b) Definieren Sie die Objektart der binären Bäume als rekursive Verbundart! Mit welchem Konzept realisiert man rekursive Verbundarten in den gängigen Programmiersprachen (PASCAL, C)?
- (3c) Beschreiben Sie eine Technik zur Umwandlung beliebiger Bäume in binäre! Zeigen Sie durch eine Plausibilitätsbetrachtung, daß das Durchlaufen des Baumes in Prä-Ordnung von dieser Umwandlung nicht berührt wird!
- (3d) Welche Konstruktionsvorschrift liegt den Artvarianten zugrunde? Erläutern Sie Ihre Antwort an einem geeigneten Beispiel! Zeigen Sie, daß die unmittelbare Verwendung des zugrundeliegenden mathematischen Konzeptes eine statische Typprüfung nicht zuläßt!

### 4. Programmiermethodik

Betrachten Sie folgende Spezifikation eines abstrakten Datentyps:

```
init:                -->  type
add:      type x nat -->  type
remove:   type       -->  type
is_empty: type       -->  boolean
first:    type       -->  nat
is_empty(init)           = true
is_empty(add(t,n))       = false
first(add(init,n))       = n
remove(add(init,n))      = init
first(add(add(t,n),m))   = first(add(t, n))
remove(add(add(t,n),m))  = add(remove(add(t,n),m))
```

- (4a) Erläutern Sie unter Verwendung dieses Beispiels, was die *wesentliche* Idee der algebraischen Spezifikation ist!
- (4b) Die angegebene Spezifikation definiert einen in der Informatik häufig anzutreffenden Datentyp. Welcher Datentyp ist das? Begründung!

- (4c) Das angegebene Gleichungssystem induziert auf der Termalgebra eine Kongruenzrelation. Geben Sie deren formale Definition an!
- (4d) Zeigen Sie, daß es in jeder der durch (c) definierten Kongruenzklassen genau einen Term gibt, der entweder die Operation *remove* überhaupt nicht oder nur in der Form *remove(init)* enthält!

## 5. Übersetzerbau

- (5a) Was versteht man unter einem Compilergenerator? Erläutern Sie Aufgabe, Aufbau und Datenfluß!
- (5b) Als Eingabe für einen Compilergenerator reicht eine kontextfreie Grammatik nicht aus. Wie werden die nichtkontextfreien Aspekte der Syntax und die Semantik beim Compilergenerator *Yacc* berücksichtigt?
- (5c) Betrachten Sie folgende Grammatik für Schleifen:

```
WHILE_stat ::= WHILE b LOOP seq_of_stat END;  
seq_of_stat ::= statement | seq_of_stat statement  
statement  ::= WHILE_stat | other_stat;
```

Verändern und/oder ergänzen Sie die erste dieser Produktionen, so daß *Yacc* die Sprunganweisungen, die zur Realisierung einer Schleife nötig sind, an der richtigen Stelle erzeugt. Begründen Sie Ihre Änderung!

- (5d) Bei Erreichen eines Syntaxfehlers kommt es darauf an, die Syntaxanalyse so fortzusetzen, daß möglichst wenige Folgefehler auftreten. Beschreiben Sie das auf Graham und Rhodes zurückgehende Verfahren zum Wiederaufsetzen! Ist das Verfahren im *Yacc* einsetzbar? (Hinweis: Bei der Beschreibung spielen die folgenden Begriffe eine Rolle: Kondensierung, Rückwärts-, Vorwärtsschritte.)