

---

**Prüfungsteilnehmer**

**Prüfungstermin**

**Einzelprüfungsnummer**

---

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

**Herbst**

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

**2002**

**66112**

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

---

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

**- Prüfungsaufgaben -**

**Fach:** Informatik (vertieft studiert)

**Einzelprüfung:** Automatentheorie, Komplexität, Algorith.

**Anzahl der gestellten Themen (Aufgaben):** 2

**Anzahl der Druckseiten dieser Vorlage:** 7

Bitte wenden!

## Thema Nr. 1

**Sämtliche Teilaufgaben sind zu bearbeiten!**

Aufgabe 1

Gegeben ist ein endlicher Automat A mit

Zustandsmenge  $S = \{S_0, S_1, S_2, S_3, S_4, S_5\}$ , Alphabet  $\Sigma = \{0, 1\}$ , Anfangszustand  $S_0$ , Endzuständen  $\{S_4, S_5\}$  und der folgenden Zustandsübergangsfunktion  $\delta$ :

$\delta(S_0, 0) = S_1, \delta(S_0, 1) = S_2, \delta(S_1, 0) = S_4, \delta(S_1, 1) = S_5, \delta(S_2, 0) = S_0, \delta(S_2, 1) = S_0,$   
 $\delta(S_3, 0) = S_5, \delta(S_3, 1) = S_4, \delta(S_4, 0) = S_3, \delta(S_4, 1) = S_5, \delta(S_5, 0) = S_3, \delta(S_5, 1) = S_4$

- Zeichnen Sie das Zustandsdiagramm des Automaten! Ist der Automat deterministisch?
- Welche Zustände sind äquivalent (mit Begründung)?
- Konstruieren Sie einen zu A äquivalenten, minimalen deterministischen Automaten!
- Geben Sie eine rechtslineare Grammatik an, die die Sprache  $L(A)$  (von A) erzeugt!
- Geben Sie einen regulären Ausdruck mit der Sprache  $L(A)$  an!

Aufgabe 2

Gegeben sei das Alphabet  $\Sigma = \{a, b\}$  und die Sprache  $L = \{w \in \Sigma^* \mid w \text{ enthält gleich viele } a \text{ wie } b\}$ !

- Geben Sie eine mehrdeutige, kontextfreie Grammatik G an, die die Sprache L erzeugt!
- Geben Sie für die Worte baab und abab jeweils einen Ableitungsbaum an!
- Zeigen Sie, dass die Grammatik G nicht eindeutig (ambig) ist!
- Gesucht ist eine eindeutige Grammatik  $G'$ , die dieselbe Sprache wie G hat!  
 Vervollständigen Sie dazu den folgenden Ansatz durch Hinzunahme von fünf weiteren Produktionsregeln:  
 $G' = (\{S, A, B\}, \Sigma, P', S)$  wobei  $P' = \{S \rightarrow aB, S \rightarrow bA, S \rightarrow bAS, \dots\}$

Aufgabe 3

Gegeben sei die Funktion  $p: \mathbb{N}^+ \rightarrow \mathbb{N}^+ \cup \{\perp\}$  mit  $p(n) = 2^{n \operatorname{div} 2}$  falls  $n \neq \perp$ ,  $p(\perp) = \perp$

Fortsetzung nächste Seite!

- a) Geben Sie eine rekursive Funktionsdefinition mit Kopf  $f(n: \text{nat}) \text{ nat}$ ; an, die die Funktion  $p$  berechnet! Dabei sollen nur arithmetische Grundoperationen (wie z.B. Addition, Multiplikation,...) verwendet werden.
- b) Bestimmen Sie das zur Funktionsdefinition in Teil a) gehörende Funktional  $\Phi$ ! (Geben Sie auch die Funktionalität von  $\Phi$  an!)
- c) Zeigen Sie, dass die Funktion  $p$  ein Fixpunkt des Funktionals  $\Phi$  ist!
- d) Zeigen Sie, dass die Funktionsdefinition von Teil a) terminierend ist!
- e) Geben Sie die Ordnung der Zeit- und der Speicherplatzkomplexität der in a) definierten Funktion an (mit Begründung)!
- f) Geben Sie eine iterative (nicht rekursive) Funktionsdefinition zur Berechnung von  $p$  an! (Wiederum dürfen nur arithmetische Grundoperationen verwendet werden!)
- g) Geben Sie die Ordnung der Zeit- und der Speicherplatzkomplexität der in f) definierten Funktion an (mit Begründung)!

#### Aufgabe 4

In einer Anforderungsanalyse für ein Banksystem wird der folgende Sachverhalt beschrieben:

Eine Bank hat einen Namen und sie führt Konten. Jedes Konto hat eine Kontonummer, einen Kontostand und einen Besitzer. Der Besitzer hat einen Namen und eine Kundennummer. Ein Konto ist entweder ein Sparkonto oder ein Girokonto. Ein Sparkonto hat einen Zinssatz, ein Girokonto hat einen Kreditrahmen und eine Jahresgebühr.

- a) Deklarieren Sie geeignete Klassen in Java oder in C++, die die oben beschriebenen Anforderungen widerspiegeln! Nutzen Sie dabei das Vererbungskonzept aus, wo es sinnvoll ist! Gibt es Klassen, die als abstrakt zu verstehen sind?
- b) Geben Sie für alle nicht abstrakten Klassen benutzerdefinierte Konstruktoren an mit Parametern zur Initialisierung der folgenden Werte: der Name einer Bank, die Kontonummer, der Kontostand, der Besitzer und der Zinssatz (bzw. Kreditrahmen und Jahresgebühr) eines Sparkontos (bzw. Girokontos), der Name und die Kundennummer eines Kontobesitzers!

Ergänzen Sie die Klassen um Methoden für die folgenden Aufgaben! Nutzen Sie wenn immer möglich das Vererbungskonzept aus und verwenden Sie ggf. abstrakte (bzw. virtuelle) Methoden!

- c) Auf ein Konto soll ein Betrag eingezahlt werden können und es soll ein Betrag abgeboben werden können. Soll von einem Sparkonto ein Betrag abgeboben werden, dann darf der Kontostand nicht negativ werden. Bei einer Abhebung von einem Girokonto darf der Kreditrahmen nicht überzogen werden.

Fortsetzung nächste Seite!

- d) Ein Konto kann eine Jahresabrechnung durchführen. Bei der Jahresabrechnung eines Sparkontos wird der Zinsertrag gut geschrieben, bei der Jahresabrechnung eines Girokontos wird die Jahresgebühr abgezogen (auch wenn dadurch der Kreditrahmen überzogen wird).
- e) Eine Bank kann einen Jahresabschluss durchführen. Dieser bewirkt, dass für jedes Konto der Bank eine Jahresabrechnung durchgeführt wird.
- f) Eine Bank kann ein Sparkonto eröffnen. Die Methode soll die folgenden fünf Parameter haben: den Namen und die Kundennummer des Kontobesitzers, die Kontonummer, den (anfänglichen) Kontostand und den Zinssatz des Sparkontos. Alle Parameter sind als String-Objekte oder als Werte eines Grunddatentyps zu übergeben! Das Sparkonto muss nach seiner Eröffnung in den Kontenbestand der Bank aufgenommen sein.

**Thema Nr. 2****Sämtliche Teilaufgaben sind zu bearbeiten!****Aufgabe 1** (Automaten und formale Sprachen)

Gegeben sei folgende Grammatik  $G$  über dem Alphabet  $\Sigma = \{a, b, <, >\}$

 $S \rightarrow <R>$  $R \rightarrow T$  $R \rightarrow <R>$  $T \rightarrow aU$  $U \rightarrow BU$  $U \rightarrow B$  $B \rightarrow a$  $B \rightarrow b$ 

$L(G)$  beschreibt die bezüglich einer Klammer korrekt geschachtelten Wörter!

- Geben Sie die von  $G$  erzeugte Sprache  $L(G)$  als Teilmenge von  $\Sigma^*$  an!
- Geben Sie einen geeigneten, möglichst einfachen Automaten an, der diese Sprache  $L(G)$  akzeptiert!
- $G_1$  sei die Teilgrammatik aus  $G$  mit dem Startsymbol  $T$ . Beschreiben Sie die zu  $L(G_1)$  gehörenden Wörter durch einen regulären Ausdruck!
- Ändern Sie die Sprache, so dass Wörter die bezüglich zweier verschiedener Klammern  $<a>a>$  und  $<b>b>$  korrekt geschachtelt sind, akzeptiert werden! Geben Sie eine kontextfreie Grammatik  $G_2$  an!
- Erweitern Sie den Automaten aus b), so dass die Schachtelung mit allgemeinen Klammern  $<w>w>$  endlicher Länge  $k$  d.h.  $|w| < k$ , analysiert werden kann. Dabei ist  $w \in L(G_1)$ . Es soll also die öffnende und die schließende Klammer durch dasselbe Wort markiert werden!
- Können Sie eine kontextfreie Grammatik angeben, die Klammern beliebiger Länge erlaubt? Wie sehen öffnende und schließende Klammern aus?

Fortsetzung nächste Seite!

**Aufgabe 2** Algorithmen

Die Fibonaccizahlen sind nach (F) definiert

$$\begin{aligned} f_0 &= 1 \\ (F) \quad f_1 &= 1 \\ f_{i+2} &= f_{i+1} + f_i \quad \text{für } i \geq 0 \end{aligned}$$

- a) Geben Sie einen funktionalen, rekursiven Algorithmus an, der die Formel (F) direkt umsetzt!
- b) Bestimmen Sie die Komplexität dieses Algorithmus gemessen an der Zahl der Additionen zur Berechnung von  $f_n$ !
- c) Geben Sie einen funktionalen Algorithmus an, der mit  $n$  Additionen  $F$  berechnet!
- d) Implementieren Sie diesen Algorithmus in Java!
- e) Implementieren Sie einen imperativen Algorithmus in Java, der  $f_n$  mit  $n - 1$  Additionen berechnet!

**Aufgabe 3** Datenstrukturen

- a) Erzeugen Sie aus der gegebenen Folge einen 2-3-4 Baum:  
22; 10; 19; 1; 13; 12; 7; 8; 5; 42; 33; 21

Fügen Sie dazu die einzelnen Elemente in gegebener Reihenfolge in einen anfangs leeren 2-3-4 Baum ein! Stellen Sie für jeden Wert die entsprechenden Zwischenergebnisse und die angewendeten Operationen als Bäume dar!

- b) In dem Ergebnisbaum suchen wir nun den Wert 17. Stellen Sie den Ablauf des Suchalgorithmus an einer eigenen Zeichnung grafisch dar!
- c) Löschen Sie nun, unter Anwendung des Algorithmus „Löschen mit Vorschau“ die Elemente mit den Werten 5, 22, 10 in der gegebenen Reihenfolge. Stellen Sie für jeden Wert die entsprechenden Zwischenergebnisse und die angewendeten Operationen als Bäume dar.

**Aufgabe 4** Algorithmen

In einem Sackbahnhof mit drei Gleisen befinden sich in den Gleisen S1 und S2 zwei Züge jeweils mit Waggons für Zielbahnhof A und B. Gleis S3 ist leer. Stellen Sie die Züge zusammen, die nur Waggons für einen Zielbahnhof enthalten! Betrachten Sie S1, S2 und S3 als Stapel und entwerfen Sie einen Algorithmus, der die Züge so umordnet, dass anschließend alle Waggons für A in S1 und alle Waggons für B in S2 stehen!

Fortsetzung nächste Seite!

**Aufgabe 5** Objektorientierter Entwurf

Sie sollen in dieser Aufgabe eine objektorientierte Analyse für eine Bibliothek erstellen! In dem System sollen mindestens Ausleiher, der Bibliothekar und Bücher (jedes Buch ist genau einmal, d.h. nicht mehrfach vorhanden) repräsentiert sein. Desweiteren existieren folgende Geschäftsprozesse:

1. Buch ausleihen: ein Ausleiher, dessen Name und Anschrift dem System bekannt sind, leiht ein Buch aus.
2. Buch zurückgeben: Ein Ausleiher gibt ein geliehenes Buch zurück. Hier wird auch gleich bei überzogener Frist (dieses Buches) die Zahlung für die Mahnung eingefordert.
3. Mahnungen verschicken: Der Bibliothekar überprüft bei allen entliehenen Büchern die Ausleihfrist. Falls diese abgelaufen ist, verschickt er an den Ausleiher täglich eine Mahnung.

Ein Buch soll folgende Attribute besitzen: 1) Autoren, 2) Titel, 3) Jahr, 4) ISBN, 5) Genre und 6) Schlagwörter. Genre kann dabei eine von fünf verschiedenen Möglichkeiten sein und es existieren mindestens 50 verschiedene Schlagwörter, die allerdings vom Benutzer erweitert werden können.

Als Chefentwickler müssen Sie folgende Aufgaben erledigen:

- a) Zeichnen Sie ein Analyse-Klassendiagramm für dieses Szenario!
- b) Fertigen Sie für alle Geschäftsprozesse Sequenzdiagramme an!