| Prüfungsteilnehmer | Prüfungstermin | Einzelprüfungsnummer |
|---|---|---|
| Kennzahl: _______ | **Frühjahr 2014** | **46113** |
| Kennwort: _______ | | |
| Arbeitsplatz-Nr.: _______ | | |

# Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
## — Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

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

Anzahl der Druckseiten dieser Vorlage: **7**

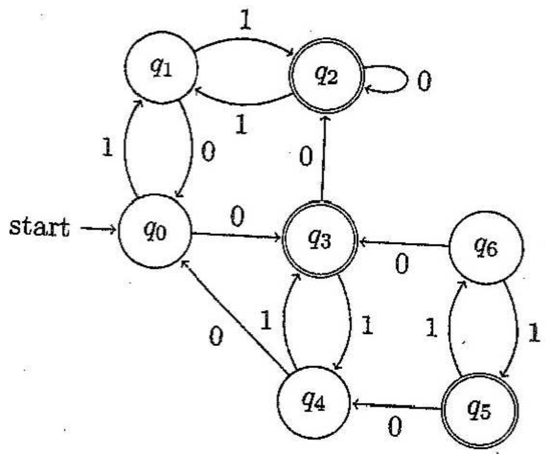
**Bitte wenden!**

## Thema Nr. 1

Aufgabe 1: Reguläre Sprachen I

Sei $\Sigma = \{a, b\}$ und $L_1$ die Sprache aller Wörter über $\Sigma$, in denen das Wort $ab$ nicht vorkommt. Sei weiterhin $L_2$ die Sprache aller Wörter über $\Sigma$, in denen $b$ genau zwei mal vorkommt.

1. Geben Sie jeweils einen regulären Ausdruck für $L_1$ und $L_2$ an.
2. Eine Grammatik heißt *rechtslinear*, wenn es für jede Produktion Nichtterminale $A$, $B$ und ein Terminalsymbol $x$ gibt, so dass die Produktion die Form $A \to xB$ oder $A \to \varepsilon$ hat. Geben Sie eine rechtslineare Grammatik für die Sprache $L_2 \setminus L_1$ an.
3. Geben Sie einen minimalen deterministischen endlichen Automaten an, der die gleiche Sprache erkennt wie der unten aufgeführte Automat! Geben Sie zu jedem Zustand des minimalen Automaten an, zu welchen Zuständen des originalen Automaten er gehört!




**Fortsetzung nächste Seite!**

Aufgabe 2: Kontextfreie Sprachen

1. Betrachten Sie die Grammatik $G = (N, \Sigma, P, S)$ mit $N = \{S, T, U, V, A, C\}$ und $\Sigma = \{a, b, c\}$. Die Menge $P$ besteht aus den folgenden Produktionen:

$$\begin{aligned} S &\to AT & \quad V &\to a \mid b \mid CU \\ T &\to UA & A &\to a \\ U &\to CV \mid AT \mid VV & C &\to c \end{aligned}$$

   Zeigen Sie mit dem CYK-Algorithmus, dass das Wort *acca* nicht in $L(G)$ liegt.

2. Zeigen Sie mit dem Pumping-Lemma für kontextfreie Sprachen, dass die folgende Sprache nicht kontextfrei ist:

   $\mathrm{L} = \{\ \mathrm{a}^n b a^n b a^n \mid n \in \mathbb{N}\}$

**Fortsetzung nächste Seite!**

Aufgabe 3: Komplexitätstheorie

Wir betrachten das Problem des Handlungsreisenden (TSP). Zur Erinnerung: Für eine Menge von Städten $S$ und eine Entfernungsfunktion $d$ zwischen diesen Städten stellen wir die folgende Frage: Existiert eine Route, die jede Stadt genau einmal besucht und dann zum Ausgangsort zurückkehrt, so dass die insgesamt zurückgelegte Entfernung höchstens $x$ ist?

Formal ist das Problem TSP wie folgt definiert:

**Gegeben:** $(S, d, x)$
$S$ ... Menge von Städten
$d$ ... Totale Distanzfunktion $d : S \times S \to \mathbb{R}$
$x$ ... Maximal erlaubte Gesamtdistanz

**Problem:** Existiert eine Folge $s_1, \ldots, s_n$ mit $|S| = n$, so dass alle $s_i$ unterschiedlich sind und die Gesamtentfernung $\left(\sum_{1 \leq i < n} d(s_i, s_{i+1})\right) + d(s_n, s_1)$ höchstens $x$ ist?

1. Begründen Sie: TSP liegt in NP.
2. Geben Sie eine polynomielle Reduktion von HAMILTON auf TSP an! Das Problem HAMILTON ist die Frage, ob ein Graph einen *Hamiltonpfad* besitzt. Ein Hamiltonpfad ist ein Weg im Graphen, der jeden Knoten genau einmal enthält.
   Formal ist das Problem HAMILTON wie folgt definiert:

   **Gegeben:** Graph $G = (E, K)$:
   $E$ ... Menge von Ecken
   $K$ ... Menge von Kanten

   **Problem:** Existiert ein Hamiltonpfad in $G$, d.h., existiert eine Folge $e_1, \ldots, e_n$ mit $|E| = n$, so dass die $e_i$ alle unterschiedlich sind und für alle $1 \leq i < n$ gilt: $(e_i, e_{i+1}) \in K$?

   *Hinweis: Wählen Sie die Distanzfunktion $d$ so, dass Sie aus der Distanz ablesen können, wie viele Kanten des Pfades nicht in $K$ liegen.*
3. Zeigen Sie: TSP ist NP-vollständig.
   *Hinweis: Sie dürfen verwenden, dass* HAMILTON *NP-hart ist.*

**Fortsetzung nächste Seite!**

Aufgabe 4: Berechen- und Entscheidbarkeit

1. Sei $\Sigma = \{0, 1\}$. Wir nehmen an, dass jedes $w \in \Sigma^*$ eine Turingmaschine kodiert und bezeichnen diese TM mit $M_w$. Die von $M_w$ berechnete Funktion bezeichnen wir mit $\varphi_w$.

   Welche der folgenden Mengen sind entscheidbar? Begründen Sie kurz!

   a) $L_1 = \{w \in \Sigma^* \mid M_w \text{ terminiert für alle Eingaben und } |w| < 10^4\}$
   b) $L_2 = \{w \in \Sigma^* \mid \varphi_w(000) = \varphi_\varepsilon(111)\}$
   c) $L_3 = \{w \in \Sigma^* \mid M_w \text{ terminiert nicht auf } 01 \text{ oder } \varphi_w(01) \in L((0+1)^*)\}$

2. Konstruieren Sie eine Turingmaschine $M$ mit Eingabealphabet $\Sigma = \{0, \#\}$, die für eine Eingabe $\#^m w$ mit $w \in \{0\}^* \setminus \{\varepsilon\}$ und $m \in \mathbb{N}$ die Ausgabe $w(\#w)^m$ liefert! Für alle anderen Eingaben darf sich $M$ beliebig verhalten.

   Beschreiben Sie auch Ihre Konstruktionsidee!

   *Beispiel:* Für die Eingabe ##00 soll $M$ die Ausgabe 00#00#00 liefern.

## Thema Nr. 2

**Annahmen:**

Sie dürfen als bekannt und bewiesen voraussetzen:
Die Sprache $\{a^n b^n \mid n \geq 1\}$ ist nicht regulär.
Die Sprache $\{a^n b^n c^n \mid n \geq 1\}$ ist nicht kontextfrei.
Um zu zeigen, dass eine Sprache L regulär (kontextfrei) ist, reicht die Angabe einer entsprechenden Beschreibung (Automat, Grammatik, Ausdruck).
Sie müssen nicht mehr zeigen, dass Ihre Beschreibung korrekt ist und genau die vorgegebene Sprache beschreibt.

**Aufgabe 1:** reguläre Mengen

Sei L die Menge aller Worte über dem Alphabet {a,b}, bei denen das zweite und letzte Zeichen übereinstimmen.

a) Geben Sie alle Worte bis zur (einschließlich) Länge zwei an!
b) Beschreiben Sie L
  i) durch einen regulären Ausdruck
  ii) durch einen deterministischen endlichen Automaten A!

**Aufgabe 2:** regulär oder nicht

Sei L die Sprache $L = \{ a^n b^m a^n \mid n,m \geq 1\}$.
Ist L regulär oder nicht?
Begründen Sie Ihre Antwort durch die Angabe einer passenden Beschreibung für L oder führen Sie den Nachweis, dass L nicht regulär sein kann!

**Aufgabe 3:** Abschlusseigenschaften

Begründen Sie:
Wenn L und L' reguläre Sprachen sind, dann ist auch die Vereinigung $L \cup L'$ regulär.

**Aufgabe 4:** kontextfrei

Zeigen Sie, dass die Sprache L (über den Zeichen {a,b,c}) kontextfrei ist!
  $L = \{(ab)^n\ c\ (ab)^n \mid n \geq 1\}$.

**Fortsetzung nächste Seite!**

**Aufgabe 5:** Chomsky Normalform und Test

a) Konstruieren Sie zur folgenden Grammatik G = ({S, A}, {a,b}, S, P) eine äquivalente Grammatik
G' in Chomsky Normalform.
Die Produktionen von G sind:
S –> a S b S, S –> X, X –> a X b, X –> ab.

b) Zeigen Sie (durch Angabe einer Ableitung),
dass das Wort w = a a b b a a b b in L(G) bzw. L(G') liegt.

**Aufgabe 6:** Berechenbarkeit

Gegeben sei die Sprache PAL = { w c $w^{rev}$ | w $\in$ {a,b}* } über dem Alphabet {a,b,c}.
$w^{rev}$ ist w gespiegelt.

a) Konstruieren Sie eine Turingmaschine (auch Mehrband oder nichtdeterministisch), die die Sprache PAL erkennt!
Beschreiben Sie, wie Ihre Turingmaschine arbeitet!

b) Welche Zeitkomplexität in O-Notation hat Ihre Turingmaschine?

**Aufgabe 7:** Komplexität

Warum ist die Sprache SAT der erfüllbaren booleschen Ausdrücke in konjunktiver Normalform in NP?

Erläutern Sie, was man für einen Ausdruck testen muss!
Nicht gefragt ist, warum das Problem sogar NP-vollständig ist.