
Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
---------------------------	-----------------------	-----------------------------

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

**Frühjahr
2016**

66115

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

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Theoretische Informatik, Algorithmen**

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

Anzahl der Druckseiten dieser Vorlage: **9**

Bitte wenden!

Thema Nr. 1
(Aufabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

1. Reguläre Sprachen

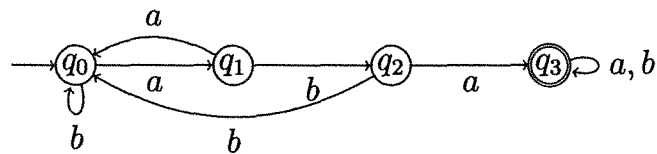
a) Geben Sie einen möglichst einfachen regulären Ausdruck für die Sprache $L_1 = \{a_1 a_2 \cdots a_n \mid n \geq 3, a_i \in \{a, b\} \text{ für alle } i = 1, \dots, n \text{ und } a_1 \neq a_n\}$ an.

b) Geben Sie einen möglichst einfachen regulären Ausdruck für die Sprache

$$L_2 = \{w \in \{a, b\}^* \mid w \text{ enthält genau ein } b \text{ und ist von ungerader Länge}\}$$

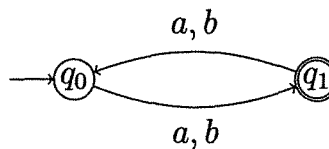
an.

c) Beschreiben Sie die Sprache des folgenden Automaten A_1



möglichst einfach und präzise in ihren eigenen Worten.

d) Betrachten Sie folgenden Automaten A_2 :



Konstruieren Sie einen endlichen Automaten, der die Schnittmenge der Sprachen $L(A_1)$ und $L(A_2)$ akzeptiert.

2. Kontextfreie Sprachen

Betrachten Sie die folgende Grammatik: $G = (V, \Sigma, S, P)$ mit

- $V = \{S, A\}$,
- $\Sigma = \{0, 1, 2\}$,
- P :

$$S \rightarrow 0S0 \mid 1S1 \mid 2A2 \mid 0 \mid 1 \mid \varepsilon$$

$$A \rightarrow A2$$

Konstruieren Sie für die Grammatik G schrittweise eine äquivalente Grammatik in Chomsky-Normalform. Geben Sie für jeden einzelnen Schritt des Verfahrens das vollständige Zwischenergebnis an und erklären Sie kurz was in dem Schritt getan wurde.

3. Klassifizierung

Im Folgenden bezeichne $\#_\sigma(w)$ die Anzahl der Vorkommen von σ in w . Sei $\mathbb{N}^+ = \{1, 2, 3, \dots\}$. Welche der folgenden Sprachen über dem Alphabet $\{a, b\}$ sind

- a) regulär?
 - b) kontextfrei, aber nicht regulär?
 - c) nicht kontextfrei?
 - d) in P?
- 1) $L_1 = \{w_1 w_2 \in \{a, b\}^* \mid w_1 \neq w_2, |w_1| = |w_2|\}$
 - 2) $L_2 = \{a^n b^n b^n a^n \mid n \in \mathbb{N}\}$
 - 3) $L_3 = \{(ab)^i a (ba)^j \mid i, j \geq 0 \text{ und } i \leq j\}$

Geben Sie zu Ihrer Lösung jeweils einen Beweis an. Erfolgt ein Beweis durch Angabe eines Automaten, so ist eine klare Beschreibung der Funktionsweise des Automaten und der Bedeutung der Zustände erforderlich. Erfolgt der Beweis durch Angabe eines regulären Ausdrucks, so ist eine intuitive Beschreibung erforderlich. Wird der Beweis durch die Angabe einer Grammatik geführt, so ist die Bedeutung der Variablen zu erläutern. Um zu zeigen, dass eine Sprache in P liegt, genügt die Angabe eines Algorithmus in Pseudocode.

4. Turingmaschinen

- a) Geben Sie eine deterministische 2-Band Turingmaschine M an, die die Funktion

$$f_M(a^n) = a^n b^n$$

berechnet. Die Maschine M nimmt somit immer einen String der Form a^n (ein String, der aus n a 's für beliebiges $n \in \mathbb{N}$ besteht) als Eingabe und produziert anschließend auf Band 2 als Ausgabe den String $a^n b^n$ (ein String aus n a 's gefolgt von n b 's).

Beschreiben Sie außerdem die Idee hinter ihrer Konstruktion.

- b) Geben Sie die Konfigurationsfolge der Turingmaschine aus (a) für die Eingabe aa an.

5. Komplexität

Das Problem k -COL ist wie folgt definiert:

Gegeben: Ein ungerichteter Graph $G = (V, E)$.

Frage: Kann man jedem Knoten v in V eine Zahl $z(v) \in \{1, \dots, k\}$ zuordnen, so dass für alle Kanten $(u_1, u_2) \in E$ gilt: $z(u_1) \neq z(u_2)$?

Zeigen Sie, dass man 3-COL in polynomieller Zeit auf 4-COL reduzieren kann. Beschreiben Sie dazu die Reduktion und zeigen Sie anschließend ihre Korrektheit.

6. Sortieren

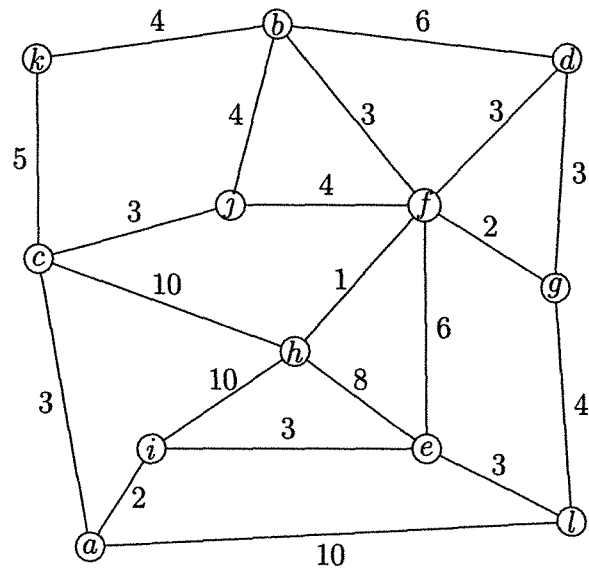
Sortieren Sie die Werte

1 45 8 53 9 2 17 10

mit Quicksort.

7. Kürzeste Pfade

Berechnen Sie mit Hilfe des Algorithmus von Dijkstra den kürzesten Pfad von Knoten a nach h .



Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

1. Verständnis formale Sprachen

Beantworten Sie kurz, präzise und mit Begründung folgende Fragen: (Die Begründungen müssen keine formellen mathematischen Beweise sein).

- a) Welche Möglichkeiten gibt es, eine formale Sprache vom Typ 3 zu definieren?
- b) Was ist die Komplexität des Wortproblems für Typ-3 Sprachen und wieso ist das so?
- c) Sind Syntaxbäume zu einer Grammatik immer eindeutig? Falls nicht, geben Sie ein Gegenbeispiel.
- d) Wie kann man die Äquivalenz zweier Typ-3 Sprachen nachweisen?
- e) Wie kann man das Wortproblem für das Komplement einer Typ-3 Sprache lösen?
- f) Weshalb gilt das Pumping-Lemma für Typ 3 Sprachen?
- g) Ist der Nachweis, dass das Typ-3 Pumping-Lemma für eine gegebene Sprache gilt, ausreichend, um zu zeigen, dass die Sprache vom Typ 3 ist? Falls nicht, geben Sie ein Gegenbeispiel, mit Begründung.
- h) Geben Sie ein Beispiel, an dem deutlich wird, dass deterministische und nicht-deterministische Typ-2 Sprachen unterschiedlich sind.
- i) Worin macht sich der Unterschied zwischen Typ 0 und 1 bemerkbar, wenn man Turingmaschinen benutzt, um das Wortproblem vom Typ 0 oder 1 zu lösen. Warum ist das so?

2. Verständnis Berechenbarkeitstheorie

Beantworten Sie kurz, präzise und mit Begründung folgende Fragen: (Die Begründungen müssen keine formellen mathematischen Beweise sein)

- a) Warum genügt es, sich auf Funktionen zu beschränken, die natürliche Zahlen auf natürliche Zahlen abbilden, wenn man untersuchen will, was heutige Computer im Prinzip berechnen können?
- b) Was besagt die Church-Turing These? Könnte man sie beweisen oder widerlegen?
- c) Für reelle Zahlen, wie z.B. π , lässt sich die Dezimaldarstellung durch entsprechende Programme beliebig genau approximieren. Gilt das für alle reellen Zahlen, d.h. lässt sich für jede reelle Zahl die Dezimaldarstellung mit entsprechenden Programmen beliebig genau approximieren?

Fortsetzung nächste Seite!

- d) Was ist für die Berechnungskraft der wesentliche Unterschied zwischen While-Berechenbarkeit und Loop-Berechenbarkeit.
- e) Die Ackermannfunktion ist ein Beispiel einer totalen Funktion, die While-berechenbar, aber nicht Loop-berechenbar ist. Sie verallgemeinert die Idee, dass Multiplikation die wiederholte Addition ist, Exponentiation die wiederholte Multiplikation, Hyperexponentiation die wiederholte Exponentiation usw. Die Stufe dieser hyper-hyper ... Exponentiation ist ein Parameter der Ackermannfunktion. Generieren Sie aus dieser Idee ein Argument, das illustriert, warum die Ackermannfunktion nicht Loop-berechenbar ist.
- f) Geben Sie ein Beispiel einer Menge an, die abzählbar, aber nicht rekursiv aufzählbar ist, und begründen Sie es.
- g) Wie ist der Zusammenhang zwischen rekursiv aufzählbar und semi-entscheidbar?

3. Verständnis Komplexitätstheorie

Beantworten Sie kurz, präzise und mit Begründung folgende Fragen: (Die Begründungen müssen keine formellen mathematischen Beweise sein)

- a) In der O-Notation insbesondere für die Zeitkomplexität von Algorithmen lässt man i.A. konstante Faktoren oder kleinere Terme weg. Z.B. schreibt man anstelle $O(3n_2 + 5)$ einfach nur $O(n_2)$. Warum macht man das so?
- b) Was ist die typische Vorgehensweise, wenn man für ein neues Problem die NP-Vollständigkeit untersuchen will?
- c) Was könnte man tun, um $P=NP$ zu beweisen?
- d) Sind NP-vollständige Problem mit Loop-Programmen lösbar? (Antwort mit Begründung!)
- e) Wie zeigt man aus der NP-Härte des SAT-Problems die NP-Härte des 3SAT-Problems? (3SAT ist ein SAT-Problem wobei alle Klauseln maximal 3 Literale haben.)

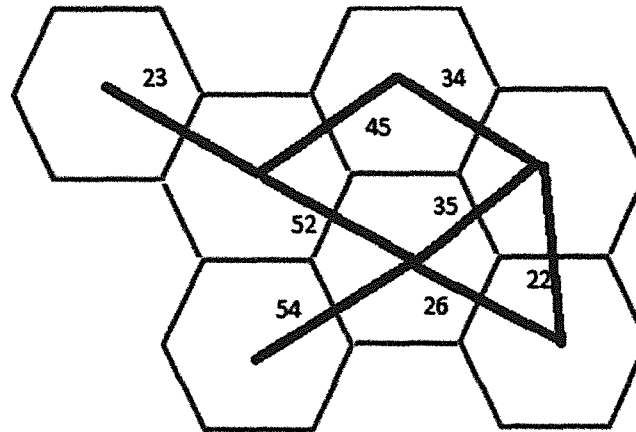
4. Hashing

Betrachte eine Hashtabelle der Größe $m = 10$.

- a) Welche der folgenden Hashfunktionen ist für Hashing mit verketteten Listen am besten geeignet? Begründen Sie Ihre Wahl.
 - 1. $h_1(x) = (4x+3) \bmod m$
 - 2. $h_2(x) = (3x+3) \bmod m$.
- b) Welche der folgenden Hashfunktionen ist für Hashing mit *offener Adressierung* am besten geeignet. Begründen Sie Ihre Wahl.
 - 1. $h_1(x, i) = (7x+i \cdot m) \bmod m$
 - 2. $h_2(x, i) = (7x+i \cdot (m-1)) \bmod m$.

5. Minimaler Spannbaum

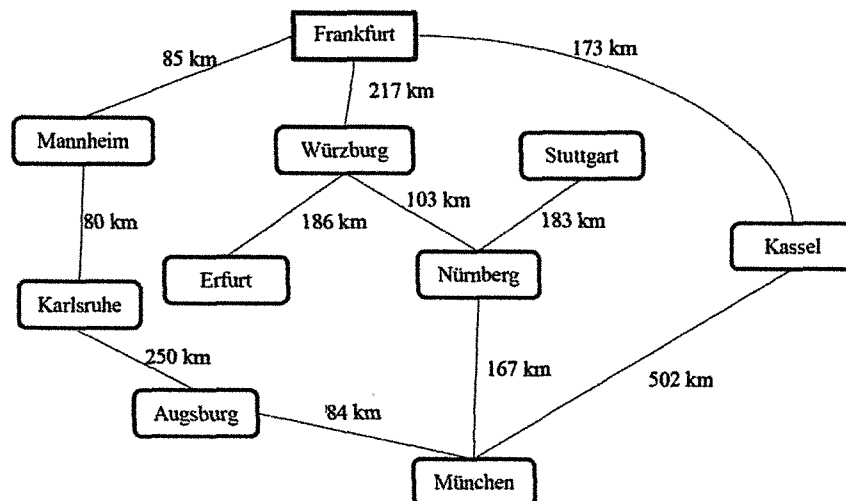
Eine Schule möchte alle seine Räume mit Internetanschlüssen versehen, hat aber wenig Geld. Sie haben sich bereit erklärt, eine Verkabelung mit möglichst wenig Kabeln zu planen. Dabei können Sie vorhandene Kabelkanäle nutzen. Der schematische Raumplan mit den ebenfalls schematisch eingezeichneten Kabelkanälen sieht folgendermaßen aus (Maße in Dezimetern).



- Beschreiben Sie das Verfahren, nach dem Sie die optimale Verkabelung wählen.
- Welche Komplexität hat das Verfahren? Woher rührt die Komplexität?
- Zeichnen Sie den Graphen mit der minimalen Verkabelung.
- Wieviel Meter Kabel benötigen Sie?

6. Dijkstra Algorithmus

- a) Berechnen Sie für folgenden Graphen den kürzesten Weg von Karlsruhe nach Kassel und dokumentieren Sie den Berechnungsweg:



- b) Könnte man den Dijkstra Algorithmus auch benutzen, um das Travelling-Salesman Problem zu lösen?

7. Verständnis Suchbäume

Wofür eignen sich die folgenden Baum-Datenstrukturen im Vergleich zu den anderen angeführten Baumstrukturen am besten, und warum. Sprechen Sie auch die Komplexität der wesentlichen Operationen und die Art der Speicherung an.

- a) Rot-Schwarz Baum
- b) AVL-Baum
- c) Binärer-Heap
- d) B-Baum
- e) R-Baum