

Informatik D: Einführung in die Theoretische Informatik
Klausur — SoSe 2018 — 08. Oktober 2018

Nebentermin, Prüfungsnr. 1007049

Gruppe: Nuss/Beere

Unbedingt ausfüllen

Matrikelnummer	Studiengang/Abschluss	Fachsemester
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>
Nachname	Vorname	
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	
Unterschrift	Identifikator <small>(Beliebiges Wort zur Identifikation im anonymen Notenaushang)</small>	
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	

Grundregeln

- Die Bearbeitungszeit der Klausur beträgt **120 Minuten**.
- Sie schreiben diese Klausur **vorbehaltlich** der Erfüllung der **Zulassungsvoraussetzung**. Das heißt: Wir werden Ihre Zulassung vor Korrektur prüfen; die Tatsache, dass Sie die Klausur mitschreiben, bedeutet keine implizite Zulassung.
- Es sind **keine Unterlagen** und auch **keine** anderen **Hilfsmittel** erlaubt.
- Benutzen Sie nur dokumentenechten (blauen oder zur Not schwarzen) **Kugelschreiber!** Bleistiftlösungen werden nicht gewertet!
- Es zählt die Antwort, die sich im dafür vorgesehenen Kästchen befindet! Soll eine andere Antwort gewertet werden, so ist diese **eindeutig** zu kennzeichnen!
- Jegliches Schummeln, und auch der Versuch desselben, führt zum Ausschluss von der Klausur und einer Bewertung mit **5,0**.

Wird vom Korrektor/Prüfer ausgefüllt

Aufgabe	1	2	3	4	5	6	7	8	9	10	Σ
Punkte (max)	16	8	12	10	8	12	14	20	16	10	126
Punkte (erreicht)											

Punkte	0..62	63..69	70..77	78..82	83..87	88..92	93..96	97..101	102..106	107..114	115..126
Note	5,0	4,0	3,7	3,3	3,0	2,7	2,3	2,0	1,7	1,3	1,0

Note:

Aufgabe 1: Sprachgrundlagen

(16 Punkte)

(a) Faktenwissen

(6 Punkte)

Welche Form hat eine Grammatik gemäß Chomsky-Hierarchie, die eine reguläre Sprache beschreibt?

Welche Eigenschaft muss der Wertebereich einer Substitutionsfunktion $h: \Sigma \rightarrow \{L_1, L_2, \dots\}$ aufweisen, um die Abgeschlossenheit regulärer Sprachen zu erhalten?

Welche Sprachfamilie wird genau durch DKAen beschrieben?

(b) Sprachen klassifizieren

(10 Punkte)

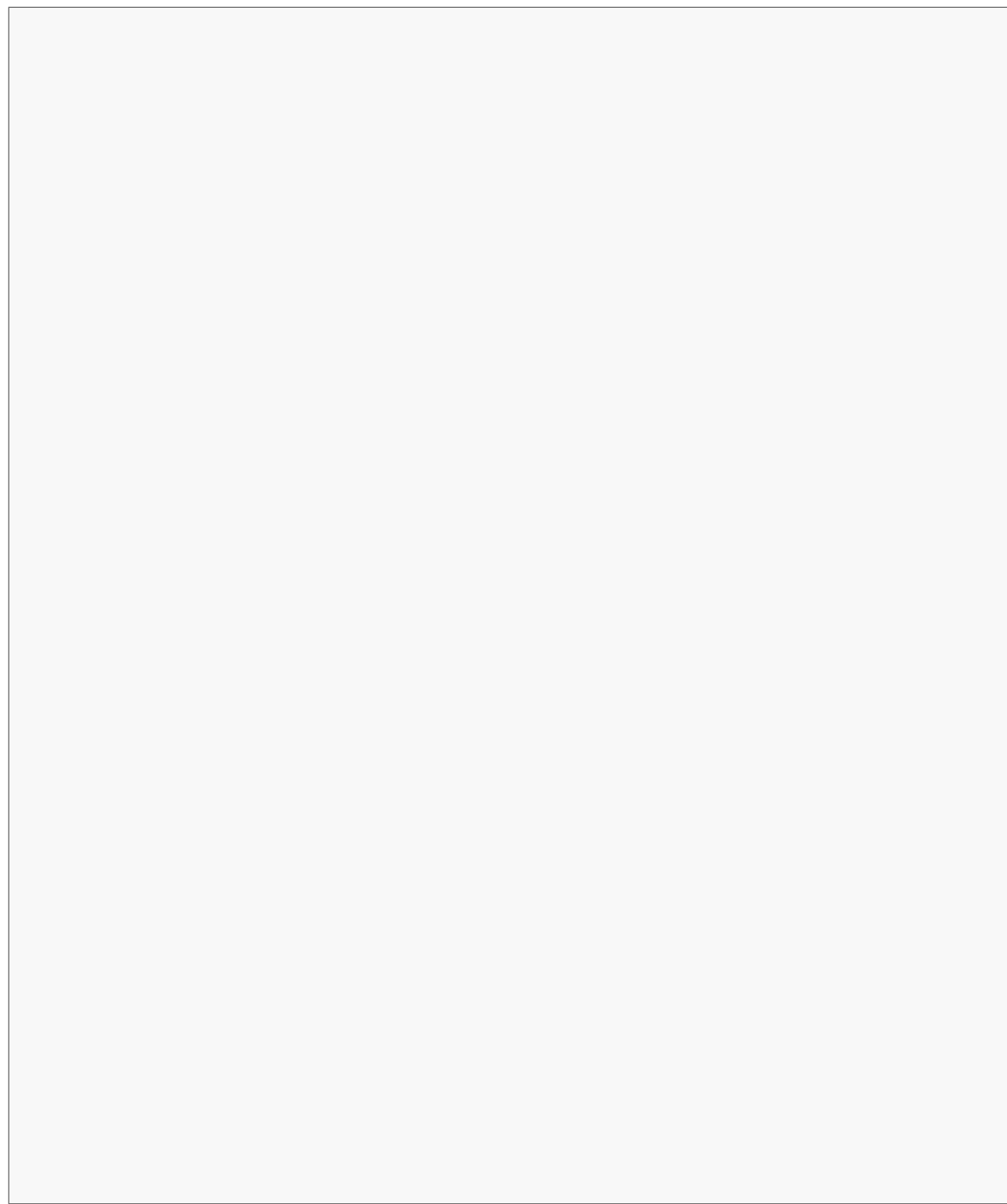
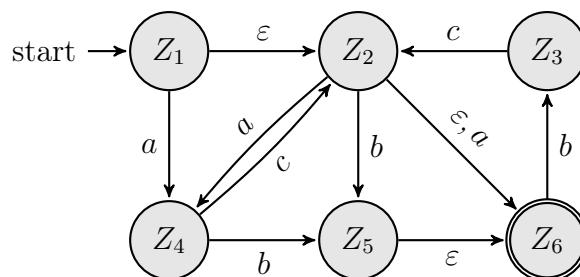
Zu welcher Sprachklasse gehören die folgenden Sprachen? Kreuzen Sie dabei *alle* korrekten Antworten an.

	regulär	determ. kontextfrei	kontextfrei	kontextsensitiv	rek. aufzählbar
$\{x^i y^j z^i \mid i \geq 0\}$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\{a^i b a b b^j c c \mid i, j \in \mathbb{N}\} \cup \mathcal{L}(1^+ 232(23 14)^*)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\{\alpha w w^R \beta \alpha \mid w \in \{\alpha, \beta\}^*\}$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\{a^n b^{2n} c^{3m} d^m a^n \mid n, m \in \mathbb{N}\} \cap \mathcal{L}(a^7 b^* c^+ d^+ a^3 a^*)$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$\left\{ w \mid \begin{array}{l} w \text{ kodiert ein GOTO-Programm, das} \\ \text{auf das Leerwort angewendet anhält} \end{array} \right\}$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Aufgabe 2: Umwandlung NDEA \rightarrow DEA

(8 Punkte)

Wandeln Sie den folgenden endlichen Automaten – gemäß dem Vorgehen aus der Vorlesung – in einen deterministischen endlichen Automaten um.



Aufgabe 3: Pumping Lemma

(12 Punkte)

(a) **Definition**

(4 Punkte)

Wie lautet das Pumping Lemma für reguläre Sprachen?

Sei L eine reguläre Sprache. Dann...

... $z = uvw$ mit den Eigenschaften

(1) , (2) und (3) .

(b) **Anwendung**

(8 Punkte)

Beweisen Sie, dass $L := \{ab^{i+j}c^j \mid i, j \in \mathbb{N}\}$ nicht regulär ist.

Aufgabe 4: Turingmaschine**(10 Punkte)**

Sei $\mathbb{T}(\alpha) \in \{0, 1, 2\}^+$ die ternäre Kodierung einer Zahl $\alpha \in \mathbb{N}$. Zum Beispiel ist $\mathbb{T}(5) = 12$.

Geben Sie eine Turingmaschine an, die für ein beliebiges $\alpha > 0$ die folgende Funktion berechnet:

$$f(\alpha) := \begin{cases} \mathbb{T}(\alpha - 1) & \alpha \text{ gerade} \\ \text{undef} & \text{sonst} \end{cases}$$

Sie dürfen folgende Aussage ohne Beweis verwenden: Eine Zahl $\alpha \in \mathbb{N}$ ist genau dann gerade, wenn die Summe der Ziffern von $\mathbb{T}(\alpha)$ (ihre *Quersumme zur Basis 3*) gerade ist.

Aufgabe 5: Turing-Vollständigkeit

(8 Punkte)

Betrachten Sie die folgende Programmiersprache ARROWS:

Variablenbezeichner sind beliebige Wörter aus $\{a, b, \dots, z, 0, 1, \dots, 9\}^+$. Die Variablen nehmen natürliche Zahlen als Werte an. Mit c werden beliebige Konstanten aus \mathbb{N} bezeichnet. Es gibt die folgenden ARROWS-Anweisungen:

- $pf \leftarrow c$ weist der Variablen pf den Wert c zu,
- $pf \uparrow c$ erhöht den Wert der Variablen pf um c ,
- $pf \downarrow c$ verringert den Wert der Variablen pf um c ,
- $pf \leftarrow eil$ weist der Variablen pf den Wert der Variablen eil zu,
- $pf \uparrow eil$ erhöht den Wert der Variablen pf um den Wert der Variablen eil ,
- $pf \downarrow eil$ verringert den Wert der Variablen pf um den Wert der Variablen eil ,
- $\mathcal{A} \Leftrightarrow pf$ führt Anweisung \mathcal{A} nur aus, wenn der Wert der Variablen pf genau 0 ist,
- $\looparrowright \dots \loopleft$ führt die Anweisungen zwischen den Symbolen in **umgekehrter** Reihenfolge in einer Endlosschleife aus,
- \curvearrowright springt aus dem innersten durch \looparrowright und \loopleft definierten Block heraus oder beendet das Programm, falls keine entsprechenden umgebenden Symbole existieren.

Eine Verkettung von Anweisungen findet durch Leerzeichen oder Zeilenumbruch zwischen den Anweisungen statt.

Beispiel: Sei in i ein Wert > 0 gespeichert. Das folgende ARROWS-Programm setzt fb auf die i -te Fibonaccizahl:

$fb \leftarrow 1 \quad i \downarrow 1 \quad \curvearrowright \loopleft i \quad b \leftarrow 0 \quad \looparrowright \quad \curvearrowright \loopleft i \quad i \downarrow 1 \quad fb \uparrow a \quad b \leftarrow fb \quad a \leftarrow b \quad \loopleft$

Zeigen oder widerlegen Sie, dass ARROWS Turing-vollständig ist.

Aufgabe 6: Entscheidbarkeit

(12 Punkte)

(a) Post'sches Korrespondenzproblem

(6 Punkte)

Sei $\mathcal{T} = \{(x_i, y_i)\}_{1 \leq i \leq k}$ eine Instanz des Post'schen Korrespondenzproblems (PCP). Wie sieht ein JA-Zeuge für \mathcal{T} aus, der sich in Linearzeit verifizieren lässt?

Betrachten Sie den folgenden nichtdeterministischen Algorithmus zur Lösung des PCPs:

Rate einen (potentiellen) JA-Zeugen.
Verifiziere, ob es sich um einen JA-Zeugen handelt.
Falls ja: **return** $\mathcal{T} \in \text{PCP}$; sonst: **return** $\mathcal{T} \notin \text{PCP}$.

Dieses Vorgehen legt den Schluss nahe, dass PCP in **NP** liegt. Aus der Vorlesung wissen wir jedoch, dass PCP unentscheidbar ist. Wo liegt der Fehler in obiger Argumentation?

(b) **Co-Semi-Entscheidbarkeit**

(6 Punkte)

Betrachten Sie das wie folgt definierte Entscheidungsproblem POLYNOMIALTIMEWHILETERMINATOR:

POLYNOMIALTIMEWHILETERMINATOR

Gegeben: Ein WHILE-Programm \mathcal{A} und ein Polynom p .

Gefragt: Terminiert \mathcal{A} auf allen Wörtern $w \in \Sigma^*$ in jeweils höchstens $p(|w|)$ Schritten?

Zeigen Sie, dass POLYNOMIALTIMEWHILETERMINATOR *co-semi-entscheidbar* ist.

Aufgabe 7: Zusammenhänge Komplexitätstheorie

(14 Punkte)

Achtung: Pro Aussage gibt es 2/−1/0 Punkte bei einer richtigen/falschen/keiner Antwort! Es gibt jedoch keine negativen Punkte für die Teilaufgaben.

(a) Ankreuzen

(10 Punkte)

Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind.

Wahr Falsch

- | | | |
|--------------------------|--------------------------|--|
| <input type="checkbox"/> | <input type="checkbox"/> | NP ist abgeschlossen unter Bildung von Schnitt und Vereinigung. |
| <input type="checkbox"/> | <input type="checkbox"/> | Wenn TAUTOLOGIE polynomiell auf PARTITION reduzierbar ist, dann ist $NP = Co-NP$. |
| <input type="checkbox"/> | <input type="checkbox"/> | Falls jedes Problem A aus $PSPACE$ polynomiell auf eine Sprache L reduzierbar ist, dann ist L NP -vollständig. |
| <input type="checkbox"/> | <input type="checkbox"/> | Falls $NP \neq Co-NP$: Wenn A sowohl auf B als auch auf C poly. reduzierbar ist, dann ist B poly. auf C reduzierbar oder umgekehrt. |
| <input type="checkbox"/> | <input type="checkbox"/> | Das Co-Problem zur Entscheidungsvariante von FACTORIZATION liegt in NP . |

(b) Beweis

(4 Punkte)

Beweisen Sie eine der oben als wahr angekreuzten Aussagen Ihrer Wahl. Kennzeichnen Sie die gewählte Aussage.

Aufgabe 8: NP-Vollständigkeit

(20 Punkte)

(a) Definition

(8 Punkte)

Definieren Sie *NP*-Vollständigkeit, ohne den Begriff „*NP*-schwer“ zu benutzen. Definieren Sie auch den Begriff der Reduktion.

(b) **Reduktion**

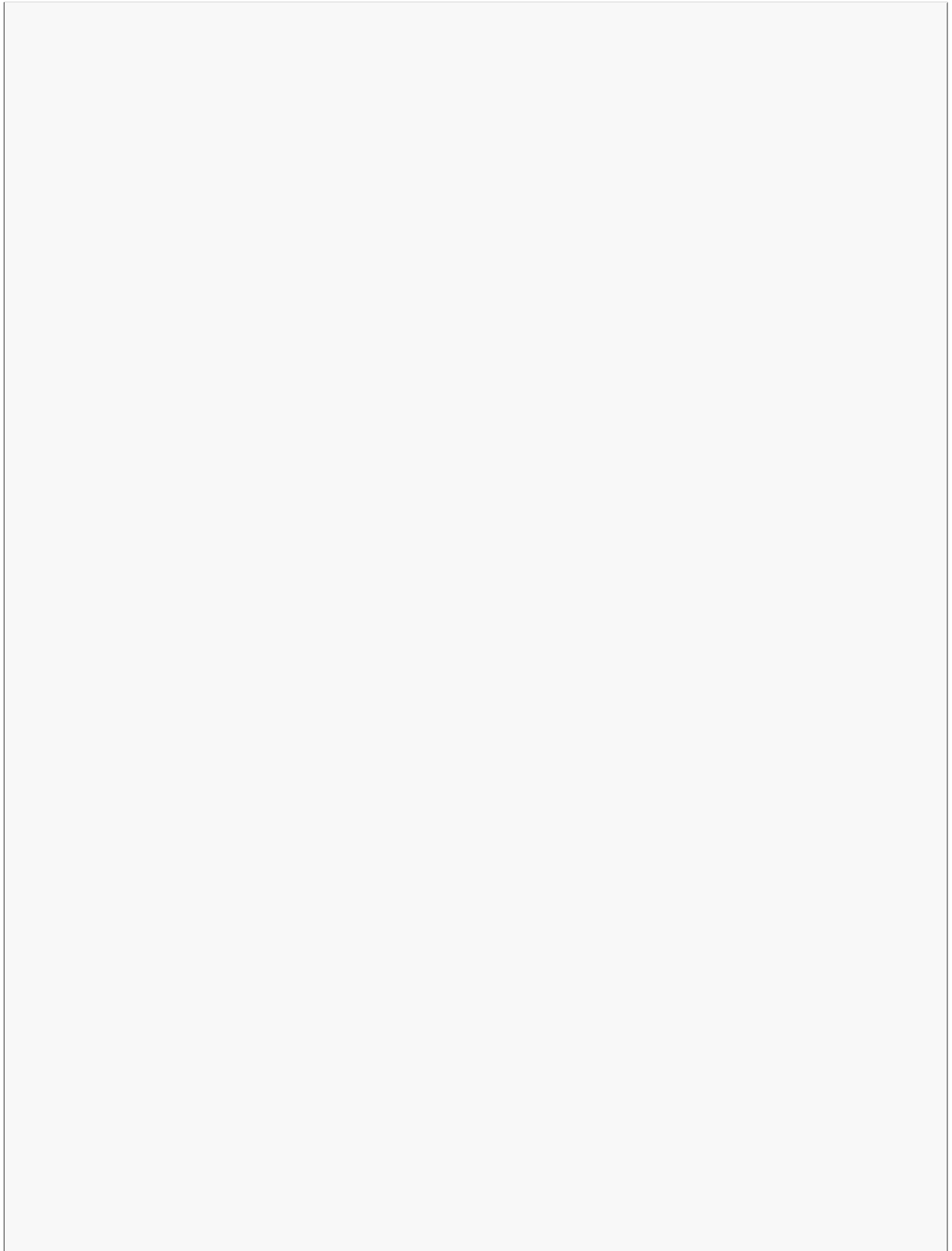
(12 Punkte)

KOCHEFFIZIENZ

Gegeben: Eine Menge von Zutaten \mathcal{Z} ; eine Menge von Rezepten \mathcal{R} , wobei jedes Rezept $r \in \mathcal{R}$ eine Teilmenge von \mathcal{Z} ist; eine natürliche Zahl $k \in \mathbb{N}$.

Gefragt: Gibt es eine Auswahl von **mindestens** k Rezepten $A \subseteq \mathcal{R}$, sodass keine Zutat mehrmals benötigt wird?

Zeigen Sie, dass das Problem KOCHEFFIZIENZ **NP**-schwer ist. Definieren Sie auch das Problem, das Sie für Ihre Reduktion benutzen.



Aufgabe 9: Dynamische Programmierung

(16 Punkte)

Sie arbeiten in einem Logistikunternehmen, das Aufträge bekommt verschiedene Güter zu transportieren. Sie wollen ihren Transportwagen mit den vorhandenen Gütern optimal packen, sodass sie nicht zu viel Gewicht beladen, aber gleichzeitig ihren Profit maximieren. Betrachten Sie das zugehörige Entscheidungsproblem:

TRANSPORTWAGEN

Gegeben: Eine Menge von Gütern $G = \{1, \dots, n\}$ mit Gewichten $w_g \in \mathbb{N}$ und Profiten $p_g \in \mathbb{N}$ für alle Güter $g \in G$, eine Gewichtsschranke W und ein mindestens zu erzielender Profit P .

Gefragt: Gibt es eine Teilmenge $G' \subseteq G$ von Gütern, sodass $w(G') := \sum_{g \in G'} w_g \leq W$ und $p(G') := \sum_{g \in G'} p_g \geq P$?

(a) Algorithmus

(10 Punkte)

Geben Sie einen deterministischen Algorithmus an, der mittels dynamischer Programmierung das Problem TRANSPORTWAGEN in Zeit $\mathcal{O}(n \cdot W)$ exakt löst. Benutzen Sie dafür ein zweidimensionales Array A mit folgender Interpretation:

$A[j, v]$ enthält den maximalen Profit einer Lösung mit Gewicht höchstens v , die nur aus den Gütern $\{1, \dots, j\}$ auswählt.

```
// Initialisierung:
```

```
// Dynamische Programmierungs-Schleife:
```

```
// Ergebnis:
```

Weiterer Platz für Aufgabe 9(a):

(b) Laufzeitanalyse & FPT-Bezug

(6 Punkte)

Handelt es sich um einen Polynomialzeitalgorithmus? Falls nicht: Um was handelt es sich dann? Begründen Sie.

Wir wollen zeigen, dass das Problem in *FPT* in Bezug auf einen Parameter k – unabhängig von der Instanzgröße – ist. Wählen Sie einen geeigneten Parameter.

Aufgabe 10: Randomisierte Algorithmen

(10 Punkte)

Sie möchten den Twitteraccount von Präsident P. Murt analysieren. Ihnen ist bekannt, dass die Tweets entweder von P. Murt persönlich oder von seiner Presseabteilung formuliert werden.

Wir nennen einen Tweet *great*, wenn er mindestens 25% der Wahlberechtigten in Rage bringt. Ein Tweet heißt *covfefe*, wenn er niemanden in Rage bringt. Ein von P. Murt formulierter Tweet ist immer *great*; stammt ein Tweet von der Presseabteilung, ist er immer *covfefe*.

Sei n die Anzahl der Wahlberechtigten. Sie verfügen über ein Programm \mathcal{P} (im folgenden *Umfrageprogramm*), das in polynomieller Zeit für einen gegebenen Tweet w und eine gegebenen Person $p \in \{1, \dots, n\}$ die Frage entscheidet: *Wird p von w in Rage gebracht?*

Betrachten Sie folgendes Problem:

TWITTERGREATNESS

Gegeben: Tweet w ; Anzahl der Wahlberechtigten n ; Umfrageprogramm \mathcal{P} .

Gefragt: Ist w great?

(a) **Algorithmus**

(6 Punkte)

Geben Sie einen **RP**-Algorithmus für TWITTERGREATNESS mit Fehlerwahrscheinlichkeit höchstens $1/2$ an.

(b) **Analyse**

(4 Punkte)

Welche Fehlerwahrscheinlichkeiten garantiert Ihr Algorithmus aus Aufgabenteil (a) in den verschiedenen Fällen? Begründen Sie, dass es sich um einen **RP**-Algorithmus handelt.

Notizen:

Notizen: