



CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

Institut für Informatik, Arbeitsgruppe Theoretische Informatik
Dr. Max A. Deppert, Janina Reuter, Annika Huch

08.06.2026

Präsenzaufgaben zur Vorlesung »Analyse von Algorithmen und Komplexität«

Blatt 9

Definition 1 (Knapsack (Entscheidungsvariante)). Gegeben ist eine Menge mit n Gegenständen mit Gewichten $w_1, \dots, w_n \in \mathbb{Z}_{\geq 0}$ und Profiten $p_1, \dots, p_n \in \mathbb{Z}_{\geq 0}$. Weiter sind eine Kapazität $K \in \mathbb{Z}_{\geq 0}$ und ein Zielprofit $P \in \mathbb{Z}_{\geq 0}$ gegeben. Bei dem KNAPSACK Problem muss entschieden werden, ob eine Teilmenge $S \subseteq [n]$ existiert, sodass

$$\sum_{i \in S} w_i \leq K \quad \text{und} \quad \sum_{i \in S} p_i \geq P$$

gilt.

Präsenzaufgabe 9.1 (KNAPSACK in NP)

Zeigen Sie $\text{KNAPSACK} \in \text{NP}$ auf zwei verschiedene Wege.

1. Geben Sie einen nicht deterministischen Algorithmus an, der das Problem in polynomialer Zeit löst.
2. Geben Sie einen polynomialen Verifizierer für das Problem an.

Wie hängen diese beiden Methoden zusammen?

Lösung. 1. Der Algorithmus arbeitet wie folgt: Für jeden Gegenstand entscheiden wir nicht-deterministisch, ob er in den Rucksack soll. Anschließend akzeptieren wir, falls die Kapazitäts- und Profitbedingung erfüllt sind.

Existiert eine valide Rucksackfüllung mit genügend Profit, dann gibt es auch eine Reihe von nichtdeterministischen Entscheidungen, die diese Füllung findet. Diese wird dann auch vom Algorithmus akzeptiert.

Der Algorithmus macht polynomial viele nichtdeterministische Entscheidungen und die Überprüfung der Eigenschaften ist ebenfalls in Polynomialzeit möglich, also ist dies ein polynomialer, nichtdeterministischer Algorithmus, der KNAPSACK löst.

2. Unser Zertifikat ist eine Teilmenge $S \subseteq [n]$, die die Auswahl für den Rucksack darstellen soll. Dies kann zum Beispiel über ein Bit pro Gegenstand realisiert werden. Um dieses Zertifikat zu verifizieren, berechnen wir $\sum_{i \in S} w_i$ und $\sum_{i \in S} p_i$ und vergleichen die Werte mit K und P und akzeptieren entsprechend.

Existiert eine valide Rucksackfüllung mit genügend Profit, dann gibt es auch ein Zertifikat, das diese Füllung beschreibt. Dieses wird dann auch vom Verifizierer akzeptiert.

Das Ausrechnen der Summen und das Überprüfen ist in polynomialer Zeit möglich, also ist dies ein polynomialer Verifizierer.

In beiden Wegen wird die extra Ressource, die wir zur Verfügung haben, genutzt, um eine Teilmenge (einen möglichen Rucksack) zu bestimmen und zu überprüfen. In der Vorlesung wird noch genau gezeigt, dass diese beiden Wege tatsächlich äquivalent sind. $\square$

Präsenzaufgabe 9.2 (Transitivität von Polynomialzeitreduktionen)

Zeigen Sie, dass Polynomialzeitreduktionen transitiv sind, d.h., dass für alle Entscheidungsprobleme $L_1, L_2, L_3 \subseteq \Sigma^*$ gilt, dass

$$L_1 \leq L_2 \wedge L_2 \leq L_3 \implies L_1 \leq L_3.$$

Lösung. Seien L_1, L_2, L_3 Sprachen. Zudem sei R_1 ein Algorithmus, der L_1 nach L_2 in $\mathcal{O}(n^a)$ reduziert, und R_2 ein Algorithmus, der L_2 nach L_3 in $\mathcal{O}(n^b)$ reduziert.

Sei x eine Instanz von L_1 der Größe n . Dann benötigt der Aufruf von R_1 auf x höchstens $\mathcal{O}(n^a)$ Zeit.

Sei x' die Ausgabe von R_1 . Dann hat x' eine Größe von höchstens $\mathcal{O}(n^a)$. Diese Schranke für die Ausgabe ergibt sich aus der Laufzeit der Reduktion. Der Aufruf von R_2 auf x' benötigt nun höchstens $\mathcal{O}((n^a)^b) = \mathcal{O}(n^{ab})$ Zeit. Damit ist die Reduktion von L_1 auf L_3 in polynomieller Zeit möglich.

Die Korrektheit ergibt sich aus der Transitivität der Äquivalenz: Sei x'' die Ausgabe von R_2 auf x' , dann gilt

$$x \in L_1 \iff x' \in L_2 \text{ und } x' \in L_2 \iff x'' \in L_3,$$

also auch $x \in L_1 \iff x'' \in L_3$. $\square$