Kapitel 4: Komplexitätstheorie

Gliederung

↓ mit minimalen Änderungen durch B. Baumgarten

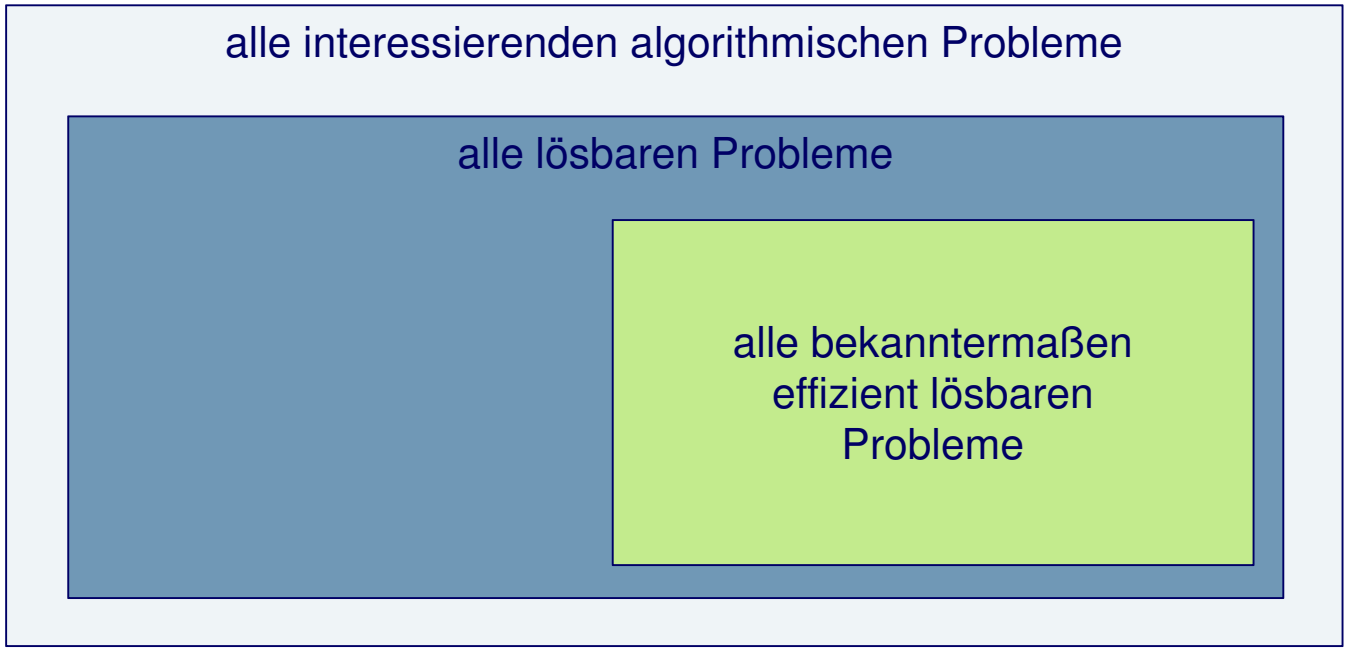
Zielstellung

- Zur Erinnerung
  - In der Komplexitätstheorie geht es darum, besser zu verstehen, dass es und warum es lösbare algorithmische Probleme gibt, die sich nicht bzw. höchstwahrscheinlich nicht effizient lösen lassen.

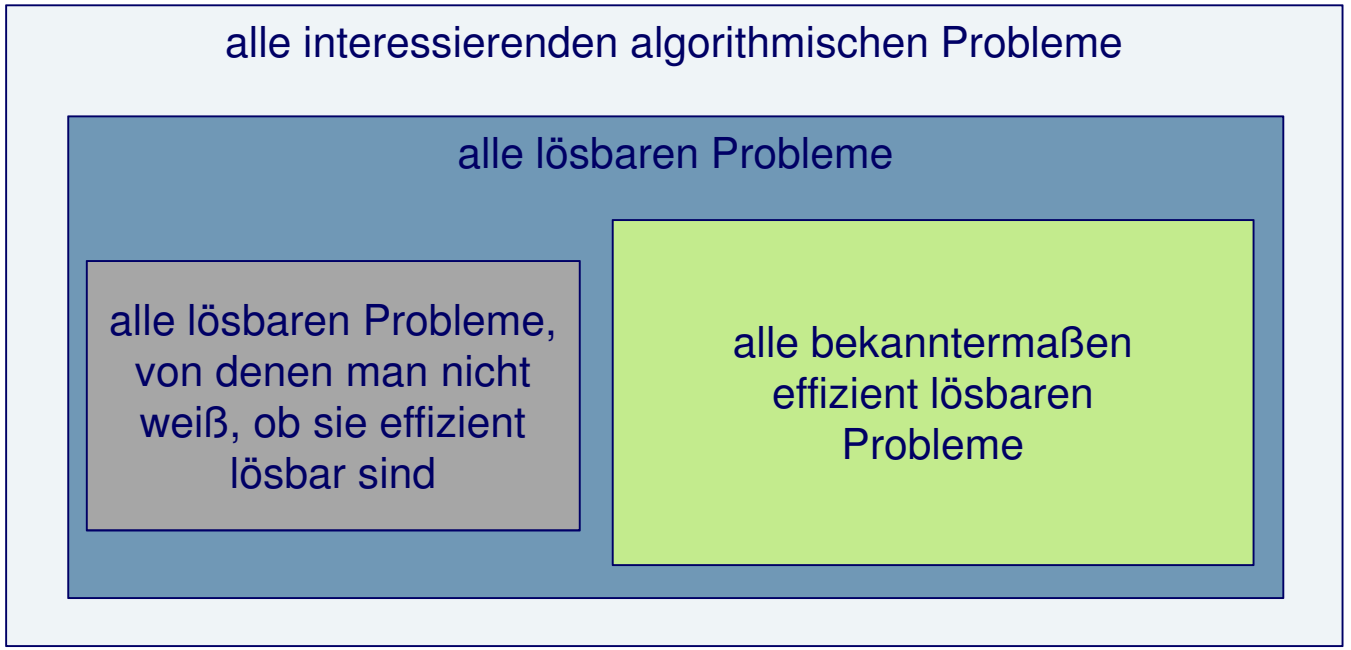
Einordnung

- prinzipielle Herangehensweise
  - Man einigt sich darauf, welcher Typ von Problemen im Fokus steht, bspw. Entscheidungs-, Konstruktions- bzw. Optimierungsprobleme, und legt damit die Klasse der jeweils interessierenden algorithmischen Probleme fest.
  - Man definiert die Klasse der Probleme, die sich mit „praxistauglichen“ Computerprogrammen lösen lassen, d.h. die Klasse der effizient lösbaren algorithmischen Probleme.
  - Man versucht, die Beziehung zwischen der Klasse der lösbaren und der Klasse der effizient lösbaren algorithmischer Probleme besser zu verstehen.

- prinzipielle Herangehensweise (cont.)
  - In der Regel wird man folgendes Gesamtbild erhalten



- prinzipielle Herangehensweise (cont.)
  - In der Regel wird man feststellen, dass es auch noch eine weitere Klasse algorithmischer Probleme gibt ...



- weiteres Vorgehen

  - Im Folgenden werden wir uns nur noch mit Entscheidungsproblemen beschäftigen.
  - Es geht also darum, besser zu verstehen, dass es und warum es Entscheidungsprobleme gibt, die sich nicht bzw. höchstwahrscheinlich nicht effizient lösen lassen.

Ein erster Grundbegriff

- Sei (X,Y) ein Entscheidungsproblem.
- Sei cod eine Funktion mit der man die Elemente von X eindeutig, und zwar möglichst platzsparend als Zeichenketten über einem Alphabet Σ beschreiben kann.
- Dann definieren wir:

> Das Entscheidungsproblem (X,Y) heißt effizient lösbar, wenn es ein „praxistaugliches“ Computerprogramm gibt, das für jedes $x \in X$ folgendes leistet:
>
> - Bei Eingabe von cod(x) wird die Ausgabe 0 bzw. 1 bestimmt;
> - wenn $x \in Y$ gilt, wird die Ausgabe 1 bestimmt;
> - wenn $x \notin Y$ gilt, wird die Ausgabe 0 bestimmt.

- Anmerkung (ein wenig formaler)
  - Sei (X,Y) ein effizient lösbares Entscheidungsproblem
  - Dann gibt es eine Turing-Maschine M, ein $k \in N_0$ und ein $c \in N$ gibt, so dass für alle $x \in X$ gilt:
    - M benötigt höchstens $c \cdot |cod(x)|^k$ Rechenschritte, um die Eingabe cod(x) zu verarbeiten

... d.h. M ist eine polynomiell zeitbeschränkte Turing-Maschine, die das Entscheidungsproblem (X,Y) löst.

Der erste zentrale Begriff: Die Komplexitätsklasse P

- Die Komplexitätsklasse P ist nun wie folgt definiert:

> Die Komplexitätsklasse P umfasst alle effizient lösbaren Entscheidungsprobleme.

Anmerkungen

Fakten zur Komplexitätsklasse P

- Zur Komplexitätsklasse P gehören bspw. die folgenden beiden Entscheidungsprobleme

  gegeben:
  - eine Liste A mit n natürlichen Zahlen
  - eine Liste B mit n natürlichen Zahlen

  gesucht:
  - die Antwort auf die Frage, ob B eine Permutation der Liste A ist
  - die Antwort auf die Frage, ob B die sortierte Version der Liste A ist

- Zur Komplexitätsklasse P gehört auch das Primzahlproblem. Das weiß man seit 2002.

Anmerkungen

- Fakten zur Komplexitätsklasse P (cont.)

  - Zur Komplexitätsklasse P gehören auch die folgenden Entscheidungsprobleme

    gegeben:
    - eine reguläre bzw. kontextfreie Sprache L über einem Alphabet Σ
    - eine Zeichenkette $w \in \Sigma^*$

    gesucht:
    - die Antwort auf die Frage, ob w ein Wort der Sprache L ist, d.h. ob $w \in L$ gilt

... das wissen wir bereits.
Um es praktisch auszunutzen, muss man aber ein wenig darauf achten, wie die Sprache L beschrieben ist.

Anmerkungen

- Einschub: Ein paar Details

  - Wir betrachten das Entscheidungsproblem (X,Y) mit
    - X enthält alle Paare (G,w), wobei G eine reguläre Grammatik und w eine Zeichenkette aus Σ^* ist
    - Y enthält alle Paare (G,w) aus X mit $w \in L(G)$.
  - Wir schauen uns zwei Lösungsalgorithmen für dieses Entscheidungsproblem an und diskutieren die Frage, ob sie belegen, dass es effizient lösbar ist

... sei cod im Folgenden wieder geeignet gewählt, um die Elemente von X zu beschreiben

- Einschub: Ein paar Details (cont.)

Ansatz 1:

Sei x = (G,w) gegeben

(1) Wir konstruieren einen nichtdeterministischen endlichen Automaten A mit L(A) = L(G).

(2) Wir konstruieren den Potenzmengenautomaten A‘ zu A, d.h. einen (deterministischen) endlichen Automaten A‘ mit L(A‘) = L(A).

(3) Wir prüfen, ob w $\in$ L(A‘) gilt.

- Bewertung:
  - (1) kann man effizient realisieren (sogar linear in |cod(x)|)
  - (2) kann man nicht effizient realisieren (exponentiell in |cod(x)|; da A‘ exponentiell größer als A sein kann)

... kein effizienter Lösungsalgorithmus !!!

- Einschub: ein paar Details (cont.)

Ansatz 2:

Sei x = (G,w) gegeben

(1) Wir konstruieren eine kontextfreie Grammatik G‘ in Chomsky-Normalform mit L(G‘) = L(G)
(2) Wir überprüfen mithilfe des CYK-Algorithmus, ob w $\in$ L(G‘) gilt

- Bewertung:
  - (1) kann man effizient realisieren (sogar quadratisch in |cod(x)| */))
  - (2) kann man effizient realisieren (kubisch in |cod((G‘,w))|; damit auch kubisch in |cod(x)|)

... ein effizienter Lösungsalgorithmus !!!

- Fakten zur Komplexitätsklasse P (cont.)
  - Es gibt Entscheidungsprobleme, die nachweislich nicht zur Komplexitätsklasse P gehören
    - Das liegt daran, dass man die Menge aller polynomiell zeitbeschränkten Turing-Maschinen effizient aufzählen kann.

... man kann wieder mittels Diagonalisierung beweisen, dass es ein Entscheidungsproblem gibt, dass nicht effizient lösbar ist

Fakten zur Komplexitätsklasse P (cont.)

- Es gibt einen „Graubereich“
- Man kennt einige tausend lösbare Entscheidungsprobleme mit folgenden Eigenschaften:
  - Man weiß, dass es sich um ein „effizient verifizierbares“ Entscheidungsproblem handelt;
  - man vermutet, dass das jeweilige Entscheidungsproblem nicht effizient lösbar ist.

Im Folgenden schauen wir uns einige Beispiele an …

Beispiel 1: das k-Cliquenproblem

- Über dieses Entscheidungsproblem haben wir bereits am Anfang der Vorlesung gesprochen (Stichwort: Ampelsteuerung)

gegeben:	• ein ungerichteter Graph G = (V,E) • eine Zahl $k \in N$
gesucht:	• die Antwort auf die Frage, ob es im Graphen G eine Clique der Größe k gibt

Beispiel 2: das Erfüllbarkeitsproblem

- Es geht darum herauszubekommen, ob eine aussagenlogische Formel in konjunktiver Normalform erfüllbar ist

gegeben:	• eine aussagenlogische Formel F in konjunktiver Normalform
gesucht:	• die Antwort auf die Frage, ob es eine Belegung gibt, die die Formel F erfüllt, d.h „wahr“ macht

Beispiel 3: das Partitionsproblem

- Es geht darum, zur Verfügung stehende Objekte gerecht auf zwei Personen aufzuteilen

gegeben:
- eine endliche Menge O von Objekten
- eine Funktion w, die jedem Objekt $o \in O$ seinen Wert, d.h. eine Zahl $w(o) \in \mathbb{N}$, zuordnet

gesucht:
- die Antwort auf die Frage, ob man O so in zwei disjunkte Teilmengen O_1 und O_2 zerlegen kann, dass die Objekte in beiden Teilmengen denselben Gesamtwert haben

Beispiel 4: das Teilsummenproblem

- Es geht darum, aus einer Menge von zur Verfügung stehende Objekte eine Teilmenge mit einer bestimmten Gesamtwert auszuwählen

gegeben:
- eine endliche Menge O von Objekten
- eine Funktion w, die jedem Objekt $o \in O$ seinen Wert, d.h. eine Zahl $w(o) \in N$, zuordnet
- eine natürliche Zahl b

gesucht:
- die Antwort auf die Frage, ob es eine Teilmenge O‘ von O mit dem Gesamtwert b gibt

Variante 1: *Finde die Teilmenge, deren Gesamtwert b am nächsten kommt (Rucksackproblem, mehr als nur eine Entscheidung)*

Zielstellung

- Wir versuchen, diesen „Graubereich“ besser zu verstehen.
- Dazu schauen wir uns zunächst das k-Cliquenproblem und das Partitionsproblem ein wenig genauer an, um den Begriff **effizient verifizierbares** Entscheidungsproblem besser zu verstehen.

Beobachtungen für das k-Cliquenproblem

- Sei G = (V,E) ein ungerichteter Graph,
- sei $k \in \mathbb{N}$,
- sei $V' \subseteq V$.

- Dann gilt:

> Man kann effizient überprüfen, ob V‘ eine Clique der Größe k in G ist.

Prüfe für alle $k \cdot (k-1)/2$ möglichen Kanten in V‘, ob sie vorhanden sind.

Beobachtungen für das Partitionsproblem

- Sei O eine endliche Menge von Objekten.
- Sei w eine Funktion, die jedem Objekt $o \in O$ seinen Wert, d.h. eine Zahl $w(o) \in \mathbb{N}$, zuordnet.
- Sei (O_1,O_2) eine Zerlegung von O in zwei disjunkte Teilmengen.

- Dann gilt:

> Man kann effizient überprüfen, ob die Objekte in O_1 und die in O_2 denselben Gesamtwert haben.

Addiere die endlich ($|O_1|$ bzw. $|O_2|$) vielen Werte der Objekte und vergleiche die beiden Summen.

- Wo genau liegt das Problem?

  - Wenn wir beim Gleichverteilungsproblem oder beim Cliquenproblem für jeden Lösungsversuch so leicht entscheiden können, ob er eine Lösung ist,
  - … wieso ist es dann so aufwändig, für ein vorgelegtes Problem festzustellen, ob es eine Lösung hat?

  - Nun ja: Zum einen geht es um
    - unendlich viele Problemexemplare $x \in X$, z.B. Graph G + Zahl k,
    - jedes mit einer endlichen Codierung cod(x), und
    - es gibt zu jedem vorgelegten Problemexemplar $x \in X$, z.B. (G,k),
    - exponentiell (in der Größenordnung $2^{|cod(x)|}$) viele vorlegbare Lösungsversuche, hier z.B. Teilgraphen von G.

  - Einen vorgelegten Lösungsversuch, z.B. Teilgraphen von G, können wir als „kleine Zusatzinformation“ i behandeln,
  - klein, weil |i| „nicht viel größer“ als cod(x) ist,
  - und ein erfolgreicher Lösungsversuch würde bedeuten, dass die Antwort für das Problemexemplar 1 (ja) ist.

Zwischenbilanz

- Für bestimmte Entscheidungsprobleme (X,Y) im „Graubereich“ weiß man, dass
  - es „Zusatzinformationen“ gibt, die es erlauben, die Frage zu beantworten, ob ein gegebenes $x \in X$ auch zu Y gehört,
  - man effizient überprüfen kann, ob die aktuell zur Verfügung gestellte „Zusatzinformation“ belegt, dass ein gegebenes $x \in X$ auch zu Y gehört,
  - man die „Zusatzinformationen“ mit Hilfe von „kurzen“ Zeichenketten über dem Alphabet $\Sigma = \{ 0,1 \}$ kodieren kann.

Ein zweiter Grundbegriff

- Sei (X,Y) ein Entscheidungsproblem
- Sei cod eine geeignete Funktion mit der man die Elemente von X eindeutig, und zwar möglichst platzsparend als Zeichenketten über einem Alphabet Σ beschreiben kann
- Dann definieren wir nun formal:

> Das Entscheidungsproblem (X,Y) heißt effizient verifizierbar, wenn es es ein $k \in N_0$ und ein $c \in N$ sowie ein „praxistaugliches“ Computerprogramm gibt, das für jedes $x \in X$ folgendes leistet:
>
> - Bei Eingabe von cod(x) und einem beliebigen $i \in \{ 0,1 \}^*$ wird die Ausgabe 0 bzw. 1 bestimmt;
> - wenn $x \in Y$ gilt, gibt es ein $i \in \{ 0,1 \}^*$ mit $|i| \leq c^*|cod(x)|^k$, so dass die Ausgabe 1 bestimmt wird;
> - wenn $x \notin Y$ gilt, wird für jedes $i \in \{ 0,1\}^*$ mit $|i| \leq c^*|cod(x)|^k$ die Ausgabe 0 bestimmt.

- Der zweite zentrale Begriff: Die Komplexitätsklasse NP

  - Die Komplexitätsklasse NP ist nun wie folgt definiert:

> Die Komplexitätsklasse NP umfasst alle effizient verifizierbaren Entscheidungsprobleme.

Anmerkungen

- Fakten zur Komplexitätsklasse NP

  - Sei (X,Y) ein Entscheidungsproblem.
  - Dann gilt offensichtlich:

    Wenn (X,Y) ein effizient lösbares Entscheidungsproblem ist, so ist (X,Y) auch ein effizient verifizierbares Entscheidungsproblem

    Ein einfacher Verifikationsalgorithmus ist:
    Ignoriere in (cod(x), i) den Teil i und gib das Ergebnis des Entscheidungsalgorithmus für cod(x) aus.

    ... zur Komplexitätsklasse NP gehören also insbesondere alle Entscheidungsprobleme, die auch zur Komplexitätsklasse P gehören.

- Fakten zur Komplexitätsklasse NP (cont.)
  - Es gibt Entscheidungsprobleme, die nachweislich nicht zur Komplexitätsklasse NP gehören
    - Das liegt wiederum daran, dass man die Menge aller polynomiell zeitbeschränkten Turing-Maschinen effizient aufzählen kann

Man kann mittels Diagonalisierung beweisen, dass es ein Entscheidungsproblem gibt, dass nicht effizient verifizierbar ist.

Fakten zur Komplexitätsklasse NP (cont.)

- Man kennt einige tausend „praktisch relevante“ Entscheidungsprobleme mit folgenden Eigenschaften:
  - Das jeweilige Entscheidungsproblem gehört zur Komplexitätsklasse NP;
  - man weiß nicht, ob das jeweilige Entscheidungsproblem zur Komplexitätsklasse P gehört.
- Der „Graubereich“ enthält u.a. alle diese Entscheidungsprobleme

... es gibt gut Gründe anzunehmen, dass „keines“ der Entscheidungsprobleme aus diesem „Graubereich“ effizient lösbar ist

Zusammenfassung

- Man erhält folgendes Gesamtbild