

## 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  $\mathcal{O}(3n^2 + 5)$  einfach nur  $\mathcal{O}(n^2)$ . Warum macht man das so?

Das Wachstum im Unendlich ist bestimmt durch den größten Exponenten. Konstante fälle bei einer asymptotischen Analyse weg, nicht wesentlich schneller

- (b) Was ist die typische Vorgehensweise, wenn man für ein neues Problem die NP-Vollständigkeit untersuchen will?

Die alten Probleme werden reduziert. Das neue Problem ist größer als die alten Probleme. Das Problem muss in NP liegen.

- (i) Problem *in* NP durch Angabe eines nichtdeterministischen Algorithmus in Polynomialzeit
- (ii) Problem NP-schwer via Reduktion:  $L_{\text{NP-vollständig}} \leq L_{\text{neues Problem}}$

- (c) Was könnte man tun, um  $P = NP$  zu beweisen?

Es würde genügen, zu einem einzigen NP-Problem beweisen, dass es in P liegt.

Zu einem Problem einen deterministischen Turingmaschin finden, die es in polynomineller Zeit löst.

- (d) Sind NP-vollständige Problem mit Loop-Programmen lösbar? (Antwort mit Begründung!)

nicht lösbar mit Loop-Programmen. Begründung ähnlich wie bei der Ackermann-Funktion. z. B. Passwort. Zu Passwort beliebig bräuchte man beliebige for schleifen, was dem endlichen Anzahl an Loop-Schleifen widerspricht.

- (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.)

in den Lösungen enthalten