Gliederung

Polynomzeitreduktion I

Vielleicht das wichtigste Konzept der Komplexitätstheorie!

Definition

Eine Sprache $A \subseteq \Sigma^*$ heißt **polynomiell reduzierbar auf** eine Sprache $B \subseteq \Pi^*$ (**in Zeichen** $A \leq_m^p B$), wenn es eine totale, in Polynomzeit berechenbare Funktion $f : \Sigma^* \to \Pi^*$ gibt, sodass für alle $x \in \Sigma^*$ gilt

$$x \in A \Leftrightarrow f(x) \in B.$$

Wir nennen f eine **Polynomzeit-Reduktion** von A auf B (**Beachte**: f muss weder surjektiv noch injektiv sein).

Bemerkung: „m“ in $\leq_m^p$ steht für „many-one-Reduktion”.

Mitteilungen:

(a) $A \leq_m^p B \quad \Rightarrow \quad A \leq B$

(b) $\leq_m^p$ ist transitiv, d.h. wenn $A \leq_m^p B$ und $B \leq_m^p C$, dann auch $A \leq_m^p C$
(Konkatenation der Reduktionen ist Polynomzeitreduktion von A auf C)

Transitivität der Polynomzeitreduktion

Beweis (für (b))

Sei f die Reduktionsfunktion für $A \leq_m^p B$, die in polynomieller Zeit $p(n)$ berechnet werden kann, und sei g die Reduktionsfunktion für $B \leq_m^p C$, die in polynomieller Zeit $q(n)$ berechnet werden kann.
Dann ist $g \circ f$ eine Reduktionsfunktion von A auf C, denn es gilt:

$$\forall x \in \Sigma^* : x \in A \Leftrightarrow f(x) \in B \Leftrightarrow g(f(x) \in C.$$

Die Berechnung von $f(x)$ kann in $p(|x|)$ Schritten durchgeführt werden, also gilt auch $|f(x)| \leq p(|x|)$. Daher kann $g(f(x))$ mit höchstens $q(p(|x|))$ Schritten berechnet werden. Somit ist $g \circ f$ also polynomzeitberechenbar. ■

Polynomzeitreduktion II

Lemma

Gilt $A \leq_m^p B$ und ist $B \in \mathrm{P}$ (bzw. $B \in \mathrm{NP}$), so ist auch $A \in \mathrm{P}$ (bzw. $A \in \mathrm{NP}$) .

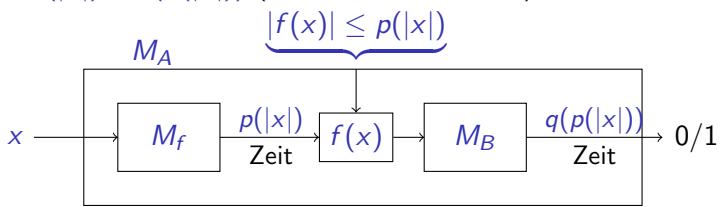
Beweis

1. $A \leq_m^p B \rightsquigarrow$ „Reduktionsfunktion“ f in $p(n)$ Schritten berechenbar durch TM M_f
2. $B \in \mathrm{P}$ (bzw. $B \in \mathrm{NP}$) $\rightsquigarrow B$ in $q(n)$ Schritten entscheidbar durch TM M_B

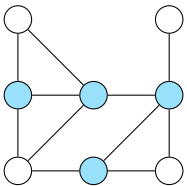
(wobei p und q Polynome)

Wie zuvor gilt $\chi_A = \chi_B \circ f$

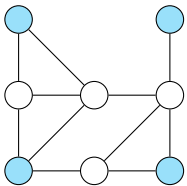
$\rightsquigarrow \chi_A$ berechnet in $p(|x|) + q(p(|x|))$ (also polynomiell viele) Schritten.



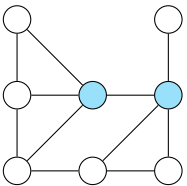
Independent Set, Vertex Cover und Dominating Set



Vertex Cover



Independent Set



Dominating Set

Eingabe: ungerichteter Graph G, Zahl $k > 0$
Frage: gibt es k Knoten in G, sodass ...

Vertex Cover: ... jede Kante in G mindestens einen dieser k Knoten als Endpunkt hat?

Independent Set: ... keine 2 dieser k Knoten mit einer Kante verbunden sind?

Dominating Set: ... jeder andere Knoten eine Kante zu mindestens einem dieser Knoten hat?

Vertex Cover und Independent Set

Theorem

Vertex Cover $\leq_m^p$ Independent Set.

Beweis

Definiere Reduktionsfunktion f vermöge $f(\langle G, k\rangle) := \langle G, |V(G)| - k\rangle$.
(offensichtlich ist f in polynomieller Zeit berechenbar)
Dann gilt:

$$\begin{aligned}\langle G, k\rangle \in \text{Vertex Cover} &\Leftrightarrow G \text{ hat eine Knotenmenge } X \subseteq V(G) \text{ mit } |X| \leq k, \text{ so dass}\\ &\quad \text{jede Kante mindestens einen Endpunkt in } X \text{ hat}\\ &\Leftrightarrow G \text{ hat eine Knotenmenge } X \subseteq V(G) \text{ mit } |X| \leq k, \text{ so dass}\\ &\quad \text{keine Kante beide Endpunkte in } V(G) \setminus X \text{ hat}\\ &\Leftrightarrow \langle G, |V(G)| - k\rangle \in \text{Independent Set}.\end{aligned}$$

NP-Vollständigkeit

Definition

Eine Sprache $A \subseteq \Sigma^*$ heißt...

a) ... **NP-schwer**, falls $\forall_{L \in \mathrm{NP}}\ L \leq_m^p A$.

b) ... **NP-vollständig**, wenn A NP-schwer ist und $A \in \mathrm{NP}$ gilt.

Anschaulich: (mit „polynomieller Unschärfe“)

1. NP-schwere Sprachen sind „mindestens so schwer“ zu entscheiden wie jede Sprache in NP
2. NP-vollständige Sprachen sind „genau so schwer“ wie jede NP-vollständige Sprache

Lemma

Ist A NP-schwer und $A \leq_m^p B$, so ist auch B NP-schwer

Beweis

Für jede Sprache $L \in \mathrm{NP}$ gilt $L \leq_m^p A \leq_m^p B$.
Somit gilt wegen Transitivität auch $L \leq_m^p B$. Also ist B auch NP-schwer.

NP-Vollständigkeit II

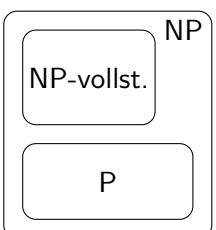
Theorem

Für jede NP-vollständige Sprache A gilt: $A \in \mathrm{P} \Leftrightarrow \mathrm{P} = \mathrm{NP}$.

Beweis

„$\Rightarrow$“: $(\forall_{L \in \mathrm{NP}}\ L \leq_m^p A) \wedge (A \in \mathrm{P}) \ \Rightarrow\ \forall_{L \in \mathrm{NP}}\ L \in \mathrm{P} \ \Rightarrow\ \mathrm{NP} = \mathrm{P}$
„$\Leftarrow$“: $(A \in \mathrm{NP}) \wedge (\mathrm{P} = \mathrm{NP}) \ \Rightarrow\ A \in \mathrm{P}$

„**Geglaubte**“ (d.h. Annahme $\mathrm{P} \neq \mathrm{NP}$) **Situation:**



Erfüllbarkeitsproblem I

SAT

Eingabe: aussagenlogische Formel F

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Beispiele

$0, 1,$ $\quad x_1, x_2, \overline{x_3},$ $\quad (x_1 \wedge \overline{x_2}),$ $\quad (\overline{(x_1 \wedge \overline{x_2})} \vee x_2 \vee \overline{x_3})$

Theorem (Satz von Cook und Levin)

SAT ist NP-vollständig.

Beweis (Idee, Details später)

Teil 1: „SAT $\in$ NP“: rate erfüllende Belegung (Zertifikat) und verifiziere sie.

Teil 2: „SAT ist NP-schwer“: mit $L \in \mathrm{NP}$ beliebig,
transformiere NTM N mit $T(N) = L$ in Formel $\varphi(x)$ sodass $x \in L \Leftrightarrow \varphi(x) \in \mathrm{SAT}$.