Grundzüge der Theoretischen Informatik

14.1.2022

Markus Bläser

Universität des Saarlandes

Kapitel 23: Reduktion und Vollständigkeit

Polynomialzeit-Reduktionen

Definition (23.1)

Seien $L, L' \subseteq \Sigma^*$.

1. $f : \Sigma^* \to \Sigma^*$ ist eine *many-one-Polynomialzeit-Reduktion* von L auf L', falls f Polynomialzeit-berechenbar ist und

   für alle $x \in \Sigma^*$ gilt: $\quad x \in L \iff f(x) \in L'$.

2. L ist *(many-one-)Polynomialzeit-reduzierbar* auf L' falls es so eine Reduktion f gibt. Wir schreiben: $L \leq_P L'$.

Polynomialzeit-Reduktionen (2)

Lemma (23.2)

Falls $\mathrm{L} \leq_{\mathrm{P}} \mathrm{L}'$ *und* $\mathrm{L}' \in \mathrm{P}$, *dann ist* $\mathrm{L} \in \mathrm{P}$.

Lemma (23.3)

$\leq_{\mathrm{P}}$ *ist transitiv.*

$$L \leq_p L' \leq_p L'' \Rightarrow L \leq L''$$

Beweis 23.3

$L \leq L' \leq L''$

F : $L \to L'$, g : $L' \to L''$

Beh.: $g \circ F$ ist eine Polynomialzeit-Red. von L auf L''

Sei $x \in \Sigma^*$ bel.

$x \in L \iff \underbrace{F(x)}_{x'} \in L' \iff \underbrace{g(F(x))}_{g(x')} \in L''$

Ber: $g \circ f$ ist Polyzeit-berechenbar

Eingabe x

1. Berechne $F(x)$ $\quad p(|x|)$
2. " $g(f(x))$ $\quad q(p(|x|))$ } polynomiell.

Da F und g Polyzeit-ber. sind gibt

es DTM M und M' und Polynome p und q,

so dass M p-zeitbeschränkt ist und F ber.

und M' q- " - " - g -"-

□

NP-Vollständigkeit

Definition (23.4)

1. L ist NP-*schwer*, falls für alle $L' \in NP$ gilt, dass $L' \leq_P L$.
2. L ist NP-*vollständig*, falls L NP-schwer ist und $L \in NP$.

"die schwierigsten Probleme in NP"

Lemma (23.5)

Falls L NP-*schwer ist und* $L \in P$, *dann ist* $P = NP$.

Lemma (23.6)

Falls L NP-*schwer ist und* $L \leq_P L'$, *dann ist* L' NP-*schwer.*

Beweis 23.5:

Sei $L' \in NP$ bel.

Da L NP-schwer ist, gilt $L' \leq_p L$.

Da $L \in P$, gilt $L' \in P$ nach Satz 23.2 $\Rightarrow NP \subseteq P$ □

Beweis 23.6.

Sei $L'' \in NP$ bel.

Da L NP-schwer ist, gilt $L'' \leq_p L$.

Nach Vor. gilt $L'' \leq_p L \leq_p L'$

Da $\leq_p$ transitiv ist gilt $L'' \leq_p L'$

Damit ist L' NP-schwer □

NP-Vollständigkeit (2)

Theorem (Cook–Karp–Levin, 23.7)

SAT *ist* NP-*vollständig.*

Lemma

23.9 Für alle $\ell \geq 3$ *ist* ℓSAT NP-*schwer.* ✓

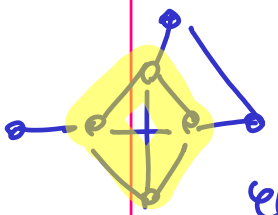
Lemma

23.10 $\ell\mathrm{SAT} \leq_{\mathrm{P}} \mathrm{Clique}$.

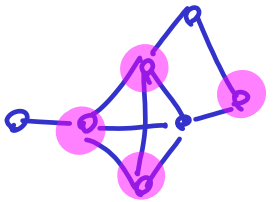
Lemma

23.11 $\mathrm{Clique} \leq_{\mathrm{P}} \mathrm{VC}$. ✓

Lemma 23.11. Clique $\leq_p$ VC



Clique



Vertex-Cover

$(G, k) \longmapsto (G', k')$

G hat evtl k-Clique $\Rightarrow$ G' hat evtl VC der Größe $\leq k'$

G hat keine k-Clique $\Rightarrow$ G' hat keines VC der Größe $\leq k'$

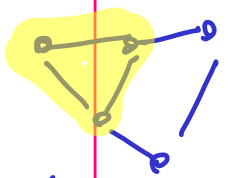
Zu einer Graph G sei $\overline{G} = (V, \binom{V}{2} \setminus E)$

"(V,E)" "Komplement graph

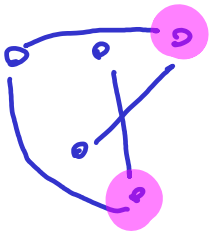
Beh.: Sei C eine Clique von G. Dann ist $V \setminus C$ ein VC von $\overline{G}$

Ist D ein VC von $\overline{G}$, dann ist $V \setminus D$ eine Clique von G

Beweis von Beh.



G



$\bar{G}$

Wenn C eine Clique von G ist, dann gilt es zwischen den Knoten von C in $\bar{G}$ keine Kanten.

D.h. jede Kante in $\bar{G}$ hat mind. einen Knoten in $V \setminus C$. Damit ist $V \setminus C$ ein VC.

Umgekehrt: Ist D ein VC in $\bar{G}$, dann gilt es zwischen Knoten in $V \setminus D$ keine Kanten.

Damit ist $V \setminus D$ eine Clique in G.

Reduktion: G' k'

$$(G, k) \longmapsto (\bar{G}, n-k)$$

Ist $(G,k) \in$ Clique, dann hat G eine k-Clique C. Damit ist $V \setminus C$ eine VC der Größe $n-k$ in $\bar{G}$. Also gilt $(\bar{G}, n-k) \in VC$

Ist umgekehrt $(\bar{G}, n-k) \in VC$, dann hat $\bar{G}$ einen VC D der Größe $n-k$.

Damit ist $V \setminus D$ eine Clique der Größe $n-(n-k) = k$. Damit ist $(G,k) \in$ Clique.

Die Reduktion ist offensichtlich Polynomialzeit-bar. $\square$

Beweis 23.9.

Wir zeigen $SAT \leq_p 3\text{-}SAT$

ϕ ist CNF $\rightarrow$ ψ ist 3-CNF

ϕ erfüllbar $\Leftrightarrow$ ψ erfüllbar

$\phi = c_1 \wedge \dots \wedge c_m$

$c_i = l_{i1} \vee l_{i2} \vee \dots \vee l_{i,k_i}$

l

Var. oder deren Negation

Variablenbelegung

$c = l_1 \vee l_2 \vee \dots \vee l_k$ (0 ∧ 0 …)

$(l_1 \vee l_2 \vee l_3) \wedge (l_4 \vee l_5 \vee l_6) \wedge \dots$ (0 ∧ 0 ∧ 0 ∧ 1 ∧ 1)

Neue Idee: Verbinde die Stücke mit neuen Var. $y_1, y_2 \dots$

$c = l_1 \vee l_2 \vee l_3 \vee \dots \vee l_k$ (0 0 ∧ … 0)

$(l_1 \vee l_2 \vee y_1) \wedge (\overline{y_1} \vee l_3 \vee y_2) \wedge \dots \wedge (\overline{y_{k-4}} \vee l_{k-2} \vee y_{k-3}) \wedge (\overline{y_{k-3}} \vee l_{k-1} \vee l_k)$ (0 0 ∧ 1 ∧ 0 0 ∧ … ∧ 1 0 0)

Jede Bel. die alle Klauseln erfüllt muss insbesondere

↳ Gilt auch auf 1 aber

Wir konstruieren ψ, indem wir für jede Klausel c wie beschrieben eine Reihe von 3-Klauseln einführen. Dabei ist es wichtig, jedes Mal neue y-Variablen zu verwenden.

z.z.: ϕ erfüllbar $\Leftrightarrow$ ψ erfüllbar.

"$\Rightarrow$" Wenn a eine erfüllende Belegung von ϕ, dann gibt es in jeder Klausel $c = l_1 \vee \dots \vee l_k$ ein Literal, das erfüllt wird. Sei l_i dieses Literal. In ψ setzen wir alle Var y_j mit $j < i-1$ auf 1 und alle anderen y_j auf 0. Dies erfüllt alle zu c gehörigen Klauseln.

"$\subseteq$" Ist umgekehrt ψ erfüllbar, dann muss

mindestens ein ursprüngliches Literal von c

erfüllt werden, denn die y_j können nur

$k-3$ Klauseln erfüllen, c wird aber

auf $k-2$ Klauseln abgebildet.

Die Abbildung $\phi \mapsto \psi$ ist offensichtlich

Polynomial-lar. $\square$

Reduktionsstrategie