Theoretische Informatik (WS04)

Prof. Dr. Raimund Seidel
Wei Ding, Dierk Johannes
Lösungsvorschlag zu Aufgabenblatt 12

Aufgabe 1

Zu zeigen ist die Transitivität von Polynomialzeitreduktionen: Aus $L \leq_p L'$ und $L' \leq_p L''$ folgt $L \leq_p L''$.

Es sei f_1 bzw. f_2 die Funktion, die $L \leq_p L'$ bzw. $L' \leq_p L''$ berechnet. Dann berechnet $f_2 \circ f_1$ eine Reduktion von L zu L'' (vgl. Übungsblatt 10, Aufgabe 1).

Zu zeigen bleibt: $f_2 \circ f_1$ ist in Polynomialzeit berechenbar. Die Funktion f_1 habe worst-case Laufzeit $c \cdot n^k$, für f_2 sei dies $d \cdot n^l$. Bei Eingabe eines Wortes w der Länge n benötigt die f_1 berechnende Turingmaschine höchstens $c \cdot n^k$ Zeit. Also hat $f_1(w)$ höchstens die Länge $c \cdot n^k$. Dieses Wort ist die Eingabe für f_2. Die Gesamtlaufzeit ist damit höchstens $d \cdot (c \cdot n^k)^l = dc^l \cdot n^{kl} =: e \cdot n^{kl}$, was ein Polynom in n ist. Also ist $f_1 \circ f_2$ in Polynomialzeit berechenbar. Insgesamt ist damit gezeigt, dass $\leq_p$ transitiv ist.

Aufgabe 2

Um zu zeigen, dass SMERF NP-vollständig ist, muss man zeigen, dass SMERF sowohl in NP ist als auch das es NP-vollständig ist.

1. SMERF ist in NP.

Errate Variablenbelegung und verifiziere sie (wie üblich).

2. SMERF ist NP-vollständig.

Dies wird gezeigt, indem wir MERF auf SMERF reduzieren. Hierzu müssen lediglich Terme der Form $Y_i \neq a$, $Y_i = Y_j$ sowie $Y_i \neq Y_j$ überführt werden in Formeln, die nur aus Termen der Form $Y_i = a$ bestehen.

$Y_i \neq a$ wird ersetzt durch:

$$\begin{cases} \mathit{false} & \text{falls } B_i = \{a\} \\ \bigvee_{b \in B_i \setminus \{a\}} (Y_i = b) & \text{sonst.} \end{cases}$$

$Y_i = Y_j$ wird ersetzt durch:

$$\begin{cases} \mathit{false} & \text{falls } B_i \cap B_j = \emptyset \\ \bigvee_{a \in B_i \cap B_j} \big((Y_i = a) \wedge (Y_j = a)\big) & \text{sonst.} \end{cases}$$

$Y_i \neq Y_j$ wird ersetzt durch: $\bigvee \big((Y_i = a) \wedge (Y_j = b)\big)$ für alle $a \in B_i, b \in B_j$ mit $a \neq b$.

Es ist zu bemerken, dass die resultierenden Formeln sehr lang werden können aber alle gestellten Bedingungen erfüllen.

Alternative: In der Vorlesung haben wir P-UNIV auf MERF reduziert. Dabei haben wir eine Formel konstruiert, die nur Terme der Formen $Y_i = a$ und $Y_i = Y_j$ beinhaltet. Es genügt also,

alle Terme der Form $Y_i = Y_j$ wie oben durch Teilformeln zu ersetzen, in denen nur Terme der Form $Y_i = a$ vorkommen. Dies reduziert P-UNIV auf SMERF.

Aufgabe 3

Was man oft vergisst, was aber trotzdem zu zeigen ist: GP liegt in NP.
Gegeben eine Instanz des GP-Problems als $m \times n$-Matrix M und als Vektor E der Länge n sowie eine (angebliche) Lösung des GP-Problems, die als Vektor L der Länge n vorliegt. Für die Lösung muss gelten: $M \cdot L \geq E$, wobei „$\geq$“ Komponentenweise zu verstehen ist. Im naiven Fall erfordert dies pro Ungleichung n Multplikationen, $n-1$ Additionen sowie einen Vergleich. Nimmt man alle diese Operationen als elementar (O(1)) an, kann man die Lösung in $O(m \cdot (n+(n-1)+1)) = O(m \cdot n)$ lösen. Ansonsten erhöht sich der Exponent entsprechend, da aber keine der Operationen exponentiell wird, bleibt die Klasse auf jeden Fall polynomiell.

Bleibt zu zeigen: GP ist NP-Schwer.
Dazu reduzieren wir von 3-SAT: Seien $C = \{C_1, \ldots, C_m\}$ Klauseln bestehend aus jeweils 3 Literalen, basierend auf der Variablenmenge $\{y_1, \ldots, y_n\}$. Zu $C = \{C_1, \ldots, C_m\}$ konstruieren wir in Polynomialzeit eine ganzzahlige Matrix M und einen ganzzahligen Vektor b, so dass alle Klauseln $C_1, \ldots, C_m$ genau dann gleichzeitig erfüllt sind, wenn es einen ganzzahligen Vektor x mit $Mx \geq b$ gibt. Die Matrix M hat $2n$ Spalten, indiziert mit $y_1, \overline{y_1}, \ldots, y_n, \overline{y_n}$. Wir fassen die Ungleichungen $Mx \geq b$ als Ungleichungen in den Literalen auf. Für $i = 1, \ldots, n$ haben wir die folgenden Ungleichungen:

$$\begin{aligned} y_i &\geq 0 \\ \overline{y_i} &\geq 0 \\ y_i + \overline{y_i} &\geq 1 \qquad (*) \\ -y_i - \overline{y_i} &\geq -1 \end{aligned}$$

Aufgrund der Ganzzahligkeitsforderung sind diese Ungleichungen genau dann lösbar, wenn y_i und $\overline{y_i}$ nur die Werte 0 oder 1 haben, aber nicht beide den gleichen Wert. Dies gilt $\forall i \in \{1, \ldots, n\}$. Für jede Klausel $C_j = z_{j,1} \vee z_{j,2} \vee z_{j,3}, j \in \{1, \ldots, m\}$, haben wir dann noch die Ungleichungen

$$z_{j,1} + z_{j,2} + z_{j,3} \geq 1. \qquad (**)$$

Die Lösbarkeit von (**) drückt aus, dass mindestens eines der Literale $z_{j,i}, i \in \{1, 2, 3\}$ in jeder Klausel C_j einen positiven Wert und zusammen mit (*) den Wert 1 haben soll. Die Koeefizienten der linken Seiten der Ungleichungen liefern dann die Einträge der Matrix M. Die Matrix M hat also Einträge aus der Menge $\{-1, 0, 1\}$ und hat die Dimension $(m+4n) \times 2n$. Die Definition des ganzzahligen Vektors b ist aufgrund der rechten Seiten der obigen Ungleichungen offensichtlich. Die Eingabegröße von $C = \{C_1, \ldots, C_m\}$ ist $3m$. Wegen $3m \geq n$ haben die Matrix M und der Vektor b zusammen Größe $O(m^2)$. Die Transformation ist also in Polynomialzeit möglich.

Wir müssen noch zeigen, dass $C_1, \ldots, C_m$ ganau dann gleichzeitig erfüllbar sind, wenn es einen ganzzahligen Vektor x gibt, der $Mx \geq b$ komponentenweise erfüllt.

„$\Rightarrow$“: Sind die Klauseln $C_1, \ldots, C_m$ gleichzeitig erfüllbar, so gibt es eine Belegung der Variablen mit den Booleschen Werten 0 oder 1, die $C_1, \ldots, C_m$ gleichzeitig erfüllen. Dann sind alle Ungleichungen der Form (*) oder (**) erfüllt. Die Ungleichungen (**) sind erfüllt, da in jeder Klausel mindestens ein Literal den Wert 1 bekommt und alle anderen werte nicht negativ sind.

„$\Leftarrow$“: Sind alle Ungleichungen (*) und (**) erfüllt, so erhalten wir mit (*) Lösungen $y_i, \overline{y_i} \in \{0, 1\}$ mit $y_i + \overline{y_i} = 1$. Die Ungleichungen (**) sind erfüllt, also hat in jeder Klausel mindestens ein Literal den Booleschen Wert 1. Eine Lösung des Ungleichungssystems, gegeben durch (*) und (**), liefert somit eine Belegung der Variablen $y_1, \ldots, y_n$, so dass alle Klauseln gleichzeitig erfüllt sind.

Aufgabe 4

(a) HaPf $\leq_P$ HaKr
Führe folgende Polynomzeittransformation durch:

- Füge einen neuen Knoten ein. (1 Schritt)
- Füge von jedem alten Knoten eine Kante zu dem neuen ein. (n Schritte)
- Füge von dem neuen Knoten eine Kante zu jedem alten ein. (n Schritte)

Die Transformation ist also in polynomieller Zeit ($2n+1$) durchführbar und es gilt: Hatte der ursprüngliche Graph einen Hamiltonschen Pfad, so hat der neue einen Hamiltonschen Kreis (gerade über den neuen Knoten). Hat der neue Graph einen Hamiltonkreis, so hatte der alte einen Hamitonpfad (der Kreis ohne den neuen Knoten).

(b) HaKr $\leq_P$ HaPf
Führe folgende Polynomzeittransformation durch:

- Wähle einen Knoten K im Graph. (1 Schritt)
- Füge zwei neue Knoten K_1 und K_2 ein. (2 Schritte)
- Füge für jede Kante, die in K hineingeht, eine entsprechende Kante in K_1 ein. ($\leq n$ Schritte)
- Füge für jede Kante, die aus K heraus geht, eine entsprechende Kante aus K_2 ein. ($\leq n$ Schritte)
- Lösche K mit allen zugehörigen Kanten. ($\leq 2n + 1$ Schritte)

Die Transformation ist also in polynomieller Zeit ($\leq 4n + 4$) durchführbar und es gilt: Hatte der ursprüngliche Graph einen Hamiltonschen Kreis, so hat der neue einen Hamiltonschen Pfad (der Kreis wurde an K aufgespaltet). Hat der neue Graph einen Hamiltonpfad, so hatte der alte einen Hamitonkreis (Pfad an K_1 und K_2 zusammengefügt).