Übungsblatt08 - NP-Vollständigkeit - Musterlösung

TH Mittelhessen, FB MNI, Berechenbarkeit und Komplexität, Prof. Dr. B. Just

Aufgabe 1

Bitte beweisen Sie, dass die Polynomialzeitrelation „$\leq_p$“ eine transitive Relation ist. D.h., sind A, B, C Sprachen mit $A \leq_p B$ und $B \leq_p C$, so ist auch $A \leq_p C$.

Lösung Aufgabe 1:

Sind A, B, C Sprachen mit $A \leq_p B$ und $B \leq_p C$, so gibt es Polynomialzeit-berechenbare Funktionen f_{AB} von den Instanzen von A in die Instanzen von B und f_{BC} von den Instanzen von B in die Instanzen von C mit

$a \in A \Leftrightarrow f_{AB}(a) \in B$ für alle Instanzen a von A, und

$b \in B \Leftrightarrow f_{BC}(b) \in C$ für alle Instanzen b von B.

Man überzeugt sich, dass die Hintereinanderausführung $f_{BC} \circ f_{AB}$ von den Instanzen von A in die Instanzen von C erfüllt

$a \in A \Leftrightarrow f_{BC} \circ f_{AB}(a) \in C$ für alle Instanzen a von A.

(Wer möchte, überzeugt sich davon, indem die beiden Richtungen „$\Rightarrow$“ und „$\Leftarrow$“ unabhängig voneinander bewiesen werden).

Wir zeigen nun noch, dass für zwei Polynomialzeit-berechenbare Funktionen f_1 und f_2 auch die Hintereinanderausführung $f_2 \circ f_1$ eine Polynomialzeit-berechenbare Funktion ist.

Seien dazu $z_1(n)$ bzw. $z_2(n)$ die maximalen Laufzeiten zur Berechnung von f_1 bzw. f_2 bei Inputs der Länge n bzw. m von geeigneten Algorithmen, und seien $k_1, k_2 \in \mathbb{N}$ Konstanten, sodass für hinreichend große n bzw. m gilt:

$$z_1(n) \leq n^{k_1} \quad \text{und} \quad z_2(n) \leq n^{k_2}.$$

Die Berechnung von $f_2 \circ f_1$ für einen Input der Länge n benötigt also höchstens

$$n^{k_1} + (n^{k_1})^{k_2}$$

Schritte, da der Output von f_1 höchstens die Länge $m = n^{k_1}$ haben kann. Dies ist weniger als $n^{k_1+k_2+1}$ für hinreichend große n, also ist $f_2 \circ f_1$ ebenfalls eine Polynomialzeitfunktion. ■

Aufgabe 2

Bitte beweisen Sie, dass 3-SAT NP-vollständig ist. Beweise aus der Vorlesung können dabei zitiert werden :).

Aufgabe 2

In der Vorlesung wurde bereits gezeigt:

i.) 3-SAT ist in NP, und

ii.) $\text{SAT}_{\text{CNF}} \leq_p$ 3-SAT.

Da SAT_{CNF} NP-vollständig ist, ist somit auch 3-SAT NP-vollständig. ■

Aufgabe 3

Bitte beweisen Sie, dass die folgenden Sprachen NP-vollständig sind.

a.) NAE-k-SAT (not all equal k-SAT).
Sei $k \in \mathbb{N} \setminus \{0\}$. Die Sprache NAE-k-SAT besteht aus den booleschen Formeln in konjunktiver Normalform mit höchstens k Literalen pro Klausel, die eine erfüllende Belegung der Variablen besitzen, sodass jede Klausel wahre und falsche Literale enthält.

b.) SET-SPLITTING (siehe Sipser, S. 325, Übung 7.29).
SET-SPLITTING ist die Menge der Paare (S, C), bei denen S eine endliche Menge ist, und $C = \{C_1, ..., C_k\} \subseteq \mathcal{P}(S)$ eine Menge von Teilmengen von S ist, mit folgender Eigenschaft: Die Elemente von S können rot oder blau eingefärbt werden, sodass keine der Mengen $C_1, ..., C_k$ nur rote oder nur blaue Elemente enthält.

Lösung Aufgabe 3:

a.)

i.) Offenbar gilt NAE-k-SAT $\in$ NP. Die nichtdeterministische TM, die die Sprache erkennt, prüft zunächst, ob der Input eine Instanz von NAE-k-SAT ist, und verwirft, falls das nicht der Fall ist. Dann rät sie nichtdeterministisch eine Belegung der Variablen, prüft, ob diese die Input-Formel durch die Belegung erfüllt wird, und akzeptiert, falls das der Fall ist.

ii.) Aber sorry .... NAE-k-SAT ist nicht NP-vollständig für $k = 1$ und $k = 2$.

Es ist NP-vollständig für $k = 3$, aber der Beweis ist schwierig ... googeln Sie bei Interesse „NAE-3-Sat np-complete“.

In dieser Musterlösung wird jetzt gezeigt: 3-SAT $\leq_p$ NAE-4-SAT. Daraus folgt dann die NP-Vollständigkeit von NAE-k-SAT für alle $k \geq 4$, weil 3-SAT NP-vollständig ist. Der Beweis ist immer noch trickreich, ohne Internet-Recherche wohl kaum möglich gewesen. Hier ist er:

Es sei

$$\Phi = \bigwedge_{j=1}^{s} (a_j \vee b_j \vee c_j)$$

eine Instanz für 3-SAT in den booleschen Variablen $x_1, ..., x_n$. Die a_j, b_j, c_j sind also Literale aus $\{x_1, ..., x_n, \overline{x_1}, ..., \overline{x_n}\}$.

Φ wird wie folgt auf eine Instanz $f(\Phi)$ von NAE-4-SAT in den booleschen Variablen $y_1, ..., y_n, z$ abgebildet:

i.) Bilde

$$\Phi_1 = \bigwedge_{j=1}^{s} (a_j \vee b_j \vee c_j \vee z)$$

ii.) Ersetze jedes x_i durch y_i und jedes $\overline{x_i}$ durch $\overline{y_i}$. Die resultierende Formel ist $f(\Phi)$. Sie kann offensichtlich in Polynomialzeit aus Φ berechnet werden.

Zu zeigen: $\Phi \in$ 3-SAT $\Leftrightarrow f(\Phi) \in$ NAE-4-SAT.

„$\Rightarrow$“:

Ist $B \in \{0,1\}^n$ eine erfüllende Belegung für Φ von $x_1, ..., x_n$, so ist $(B, 0)$ eine erfüllende Belegung von $y_1, ..., y_n, z)$ für $f(\Phi)$. Jede Klausel enthält also mindestens eine wahres Literal, und da z nicht wahr ist, auch ein falsches. Somit ist $f(\Phi) \in$ NAE-4-SAT.

„⇐“:

Sei umgekehrt $f(\Phi) \in$ NAE-4-SAT, und $(b_1, ..., b_n, z) \in \{0,1\}^{n+1}$ eine Belegung für $(y_1, ..., y_n, z)$, die jede Klausel von $f(\Phi)$ erfüllt, und in keiner Klausel alle Literale erfüllt.

Falls in dieser Belegung $z = 0$ gilt, so ist Φ erfüllbar durch die Belegung $(b_1, ..., b_n)$ für $(x_1, ..., x_n)$, also ist Φ in 3-SAT.

Falls in dieser Belegung $z = 1$ gilt, so ist Φ erfüllbar durch die Belegung $(b_1 \oplus 1, ..., b_n \oplus 1)$ für $(x_1, ..., x_n)$. Denn in jeder Klausel von $f(\Phi)$ war mindestens ein Literal nicht erfüllt - dies muss dann eines der y gewesen sein. Also erfüllt diese Belegung in jeder Klausel von Φ mindestens ein Literal. Somit ist auch hier Φ in 3-SAT.

b.)

i.) Offenbar gilt SET-SPLITTING $\in$ NP. Die nichtdeterministische TM, die die Sprache erkennt, prüft zunächst, ob der Input eine Instanz von SET-SPLITTING ist, und verwirft, falls das nicht der Fall ist. Dann rät sie nichtdeterministisch eine Färbung der Elemente von S, prüft, ob keine der Mengen $C_1, ..., C_k$ nur rote oder nur blaue Elemente enthält, und akzeptiert, falls das nicht der Fall ist.

ii.) Wir zeigen NAE-k-SAT $\leq_p$ SET-SPLITTING:

Sei Φ eine Instanz für NAE-k-SAT in den Variablen $x_1, ..., x_n$ mit k Klauseln.

Für $f(\Phi)$ sei $S = \{x_1, ..., x_n\}$, und für jede Klausel von ϕ wird eine Menge C_j gebildet, die genau die Variablen enthält, die die Klausel enthält, $j = 1, ...k$.

Die Abbildung ist offenbar Polynomialzeit-berechenbar.

Eine Belegung einer Variable mit 1 entspricht nun einer blauen Färbung, eine Belegung mit 0 entspricht einer roten Färbung.

Man sieht dann: $\Phi \in$ NAE-k-SAT $\Leftrightarrow f(\Phi) \in$ SET-SPLITTING.

Da NAE-3-SAT NP-vollständig ist, ist auch SET-SPLITTING NP-vollständig. ■

Aufgabe 4

Bitte sehen Sie sich im englischsprachigen Wikipedia die „list of NP-complete problems“ an.

a.) Stöbern Sie ein bisschen, ob Sie Probleme finden, die Sie ohne allzu großen Aufwand verstehen können. So gewinnen Sie ein Gefühl für NP-Vollständigkeit (und für Berechnungsprobleme überhaupt :)).

b.) Bitte sehen Sie sich das „Traveling Salesman Problem“ (TSP) etwas genauer an. Es ist eines der berühmtesten NP-vollständigen Probleme, mit vielen Anwendungen in der Logistik, das auch in populärwissenschaftlichen Darstellungen immer wieder als Beispiel für ein „schweres“ Problem zitiert wird.

Lösung Aufgabe 4:

Hier gibt es keine Musterlösung :).

Aufgabe 5

Der Satz von Cook (manchmal auch Cook-Levin) besagt, dass $\mathrm{SAT}_{\mathrm{CNF}}$ eine NP-vollständige Sprache ist.
Warum hätte der Beweis des Satzes nicht funktioniert, wenn man statt der 2×3-Fenster nur 2×2-Fenster betrachtet hätte?

Lösung Aufgabe 5:

Mit 2×2-Fenstern kann man nicht in CNF übersetzen, dass im Tableau eine Zeile eine Nachfolgekonfiguration der vorherigen ist. In Konfigurationen, in denen es Nachfolgekonfigurationen mit einer Kopfbewegung nach rechts und einer nach links gibt, könnte die Folgekonfiguration zwei Felder haben, in denen ein Zustand steht.

Genaueres gerne in der Übungsgruppe - aber hier gilt es zunächst, den Beweis im Sipser nachzuvollziehen :).