Definition

Für Mengen A,B definiere
$A \Delta B \quad (A \setminus B) \quad (B \setminus A)$ (= $(A \quad B) \setminus (A \quad B)$).

Lemma 3

Falls $L \Delta L'$ endlich ist, dann ist $L \quad P$ gdw $L' \quad P$.

Beweis:

Sei T eine DTM die L in polynomieller Zeit erkennt.
Es ist leicht die endlich vielen Fälle, in denen T' sich anders verhalten muss, in den Zuständen und der Übergangsfunktion zu kodieren, so dass L(T') = L'.

Bemerkung

Lemma 3 gilt für alle Komplexitätsklassen.

Theorem 4

Falls $P \neq NP$ ist, dann gibt es ein $L^* \quad NP$ mit $L^* \quad P$ und L^* ist nicht NP-vollständig.

Beweis:

Sei L NP-vollständig und $w_0 \quad L$. (Σ^* ist nach Annahme nicht NP-vollständig).
Wir definieren G und $L^* \quad \{ w \quad L \quad w \quad G \}$.
Setzte $f(x) \quad x$ falls $x \quad G$,
$f(x) \quad w_0$ sonst.
f ist P-Reduktion von L^* nach L falls " $x \quad G$" in polynomieller Zeit entscheidbar ist. Es ist dann $L^* \quad NP$ (da $L \quad NP$).

Seien $\{L_1, L_2, L_3, ...\}$ eine Aufzählung der Sprachen in P.
$\{L'_1, L'_2, L'_3, ...\}$ eine Aufzählung der NP-vollständigen Sprachen.

Definiere für i Funktionen g_i, b_i: Σ^* wie folgt:
$g_i(n)$ "das kürzeste $w \quad L \Delta L_i$ mit $w \geq n$"
$b_i(n)$ "das kürzeste $w \quad L'_i$ mit $w \geq n$"

"$g_i(n)$ ist ein Gegenbeispiel, dass zeigt $L^* \neq L_i$." $g_i(n)$ existiert immer, da sonst $L \Delta L_i$ endlich wäre und damit nach Lemma 3 $L \quad P$ (Widerspruch zu L NP-vollständig).

"$b_i(n)$ ist ein Beispiel für $L'_i \neq L^*$." $b_i(n)$ existiert immer, da sonst L'_i endlich wäre (Widerspruch zu $P \neq NP$).

Definiere noch für n ,
$r(n) \quad \max\{ g_i(n) , b_i(n) \quad i \leq n \}$

Es gelten nun folgende Eigenschaften. Für jedes n gibt es:
- für alle $i \leq n$: $w_i \quad L \Delta L_i$ mit $n \leq w_i \leq r(n)$
- für alle $j \leq n$: $w_j \quad L'_j$ mit $n \leq w_j \leq r(n)$

Wähle eine geeignete Abbildung r' mit $r'(n) > r(n)$ und definiere
$r_o \quad 0$
$r_{i+1} \quad r'(r_i)$
und weiter
$G \quad \{ n \quad r_i \leq n \leq r_{i+1}$ für i gerade $\}$

Angenommen L^* P. Dann ist $L^* = L_i$ für ein i .
Sei n gerade mit $r_n \geq i$.
Dann gibt es ein w mit $r_n \leq |w| \leq r(r_n) \leq r_{n+1}$ (also w G) und w $L \Delta L_i$.
Da w G ist und L^* L, ist w $L^* \Delta L_i$, also $L^* \neq L_i$ und das ist ein Widerspruch.

Sei jetzt angenommen L^* sei NP-vollständig. Dann ist $L^* = L'_j$ für ein j .
Sei m ungerade mit $r_m \geq j$.
Dann gibt es ein w mit $r_m \leq |w| \leq r(r_m) \leq r_{m+1}$ (also w G) und w L'_j.
Da w G ist, ist w L^*, also $L^* \neq L_j$; Widerspruch.

Damit G brauchbar ist (d.h. { x x G } P) muss r' folgende Eigenschaft haben:
(1) Es gibt eine DTM, die bei Eingabe 1^n genau r'(n) Schritte rechnet.

Es gilt aber folgendes
Lemma: Für jede berechenbare Funktion r gibt es ein r' mit (1) und r (n) < r'(n) für alle n .

Mit (1) hat man dann folgenden Algorithmus, der „ x G“ entscheidet:
Sei T die DTM aus (1) für r'.
Input x.
Berechne $r_0, r_1, r_2, r_3, \ldots, r_k$ durch Simulation von T bis $r_{k+1} \geq |x|$. (es ist dann $r_i \leq |x|$ für $i \leq k$, also wird $O(|x|)$ Zeit zur Berechnung eines r_i benötigt)
Akzeptiere falls i gerade, sonst verwerfe.

Da für alle i $r_i < r_{i+1}$ ist und man bei der Berechnung von r_{k+1} nach $|x| + 1$ Schritten aufhören kann, ist die Laufzeit des Algorithmus $0(|x|^2)$.

Bemerkung
Falls NP ≠ co-NP ist, so sind alle L co-NP nicht NP-vollständig.

In NP co-NP liegen z.B.
- PRIM (P !)
- FAKTORISIEREN = { (n,k) n hat Primfaktor p ≤ k }
- Model-Checking für Lμ

2.4 Relativierung

Definition (Orakel-Turingmaschinen)
Deterministische (bzw. nicht-deterministische) Orakel-Turingmaschinenen sind DTM'n (bzw. NTM' n)für die folgende Zusatzbedingung gilt:
- die TM hat ein spezielles Anfrageband
- die TM hat drei spezielle Zustände: ANFRAGE, JA, NEIN

Das Verhalten der Maschine ist abhängig von einem Orakel A Σ^*. Im Zustand Anfrage geht sie über in
JA falls Anfrageband A
NEIN sonst

Bezeichnung: T^A für T Orakel-TM mit Orakel A.

Definition (Relative Komplexitätsklassen)

$P^A = \{ L \mid$ es gibt Orakel-DTM T mit $L(T^A) = L$ und $TIME_T(x) = O(x^k) \}$

$NP^A = \{ L \mid$ es gibt Orakel-NTM T mit $L(T^A) = L$ und $TIME_T(x) = O(x^k) \}$

Bemerkung

$P = P$, $NP = P$

Bemerkung (Relativierung)

Wir werden zeigen:

- es gibt Orakel A mit $P^A = NP^A$
- es gibt Orakel B mit $P^B \neq NP^B$

Man sagt, eine Beweistechnik lässt sich relativieren, wenn wenn sie auf Orakel- Komplexitätsklassen mit beliebigem Orakel anwendbar ist.

Wenn also zum Beispiel die Korrektheit eines vermeintlichen Beweises für P = NP (bzw. P ≠ NP) geprüft werden soll, dann reicht es ihn zu relativieren (d.h. zu zeigen, dass sich damit auch $P^A = NP^A$ (bzw. $P^A \neq NP^A$) für jedes Orakel A zeigen lässt) um ihn zu widerlegen.

Alle Beweistechniken die wir bisher kennengelernt haben lassen sich relativieren. Wir werden aber noch Techniken zur Trennung von Komplexitätsklassen sehen, die sich nicht relativieren lassen.