# Theoretische Informatik: Blatt 9

Abgabe bis 27. November 2015
Assistent: Sacha Krug, CHN D 42

**Linus Fessler, Markus Hauptner, Philipp Schimmelfennig**

# Aufgabe 22

(a) Wir wollen zeigen, dass NTIME($f$) unter Vereinigung abgeschlossen ist.
Seien $L_1, L_2 \in$ NTIME($f$), dann gibt es nichtdeterministische MTMs $M_1, M_2$ mit

$$L(M_1) = L_1, \quad L(M_2) = L_2 \quad \text{und} \quad \text{Time}_{M_1}(n) \in \mathcal{O}(f(n)), \quad \text{Time}_{M_2}(n) \in \mathcal{O}(f(n))$$

Wir konstruieren nun eine neue MTM $M$ mit $L(M) = L := L_1 \cup L_2$.
$M$ simuliert dazu $M_1$ und $M_2$ gleichzeitig. Sobald eine von beiden akzeptiert, akzeptiert $M$ ihre Eingabe. Falls beide verwerfen, verwirft auch $M$.

Falls nun also ein $x$ in $L_1$ oder $L_2$ ist, wird $M$ akzeptieren, da $x \in L$.
Für die Berechnung braucht $M$ das Minimum der Rechenzeit beider MTMs.

$$\text{Time}_M(x) = \min\{\text{Time}_{M_1}(x), \text{Time}_{M_2}(x)\} \quad \text{für alle x.}$$

Daher:

$$\text{Time}_M(n) \in \mathcal{O}(f(n)) \quad \text{und} \quad L(M) = L \in \text{NTIME}(f)$$

(b) Wir wissen $L \in$ NTIME($f$) und $L' \in$ TIME($f$).
Es gibt also eine N-MTM $M_1$ mit $L(M_1) = L$ und eine MTM $M_2$ mit $L(M_2) = L'$.
Um zu zeigen, dass $L - L' \in$ NTIME($f$) konstruieren wir eine N-MTM $M$, die folgendermaßen funktioniert:
$M$ simuliert $M_1$ auf der Eingabe. Falls $M_1$ nicht akzeptiert, akzeptiert auch $M$ nicht. Akzeptiert $M_1$ doch, dann simulieren wir die Eingabe auch auf $M_2$. Akzeptiert $M_2$ verwerfen wir, verwirft $M_2$ akzeptieren wir.

Offensichtlich ist $L(M) = L - L'$.
Mit Hilfe von *Lemma 6.5* folgt: Das Simulieren von $M_1$ und $M_2$ liegt in $\mathcal{O}(f(n))$.
Damit liegt auch die Summe der Laufzeiten in $\mathcal{O}(n)$ und $L(M) = L - L' \in$ NTIME($f$).

# Aufgabe 23

b) Sei $L \in$ NSPACE($f(n)$) $\cap$ NTIME($f(n)^k$). Somit gibt es eine NMTM $M$ mit $L(M) = M$ wobei $\mathit{Space}_M(n) \leq c \cdot f(n)$ für ein gewisses $c$. Ausserdem folgt, dass $M$ ein Wort $w \in L(M)$ innerhalb von $c' \cdot f(n)^k$ Schritten akzeptieren muss, für ein bestimmtes $c$.
Offensichtlich können wir $c'$ mit $\mathcal{O}(1)$ Speicherplatz berechnen und speichern. $c' \cdot f(n)^k$ können wir ebenfalls offensichtlich mit $\mathcal{O}(f(n)^k)$ Speicherplatz speichern. Um nun herauszufinden, ob eine akzeptierende Konfiguration in $c' \cdot f(n)^k$ Schritten erreicht werden kann, übergeben wir der uns bekannten Funktion REACHABLE als Parameter $m$ $c' \cdot f(n)^k$.
Insofern nicht anders erwähnt, funktioniert die vorgeschlagene NMTM gleich wie in Satz 6.7.
Für das Speichern einer inneren Konfiguration wird $\mathcal{O}(f(n))$ Speicher benötigt. Die Anzahl der rekursiven Aufrufe von REACHABLE ist höchstens $\log_2(2 \cdot c' \cdot f(n)^k) = c'' * \log(f(n))$. Insgesamt ergibt dies also $\mathcal{O}(f(n)) \cdot \mathcal{O}(\log(f(n))) = \mathcal{O}(f(n) \cdot \log(f(n)))$ viel Speicherplatz.

# Aufgabe 24

Wir wissen, für jedes $w \in L$ gibt es einen Zeugen $x$, mit $|x| \leq \log_2 |w|$.
Es folgt, für jedes $w$ gibt es

$$\sum_{i=0}^{\log_2 |w|} = 2^{\log_2 |w|+1} - 1 = 2(2^{\log_2 |w|}) - 1 = 2\,|w| - 1$$

mögliche Kandidaten für einen Zeugen in $\{0,1\}^*$, falls das leere Wort ein Zeuge sein kann.
Da $A$ ein ein Polynomzeit-Verifizierer für $L$ ist, kann $A$ für gegebene $w$ und $x$ in $\mathcal{O}(|w|^k),\ k \in \mathbb{N}$ herausfinden, ob $w \in L$.
Wir testen einfach für alle möglichen $2|w| - 1$ Zeugen $x$ ob $(w, x) \in \mathrm{L}(A)$. Das hat eine Laufzeit von $\mathcal{O}((2|w| - 1) \cdot |w|^k) \subset \mathcal{O}(|w|^{k'})$.
Da wir also für alle Wörter $w$ in polynomieller Zeit feststellen können, ob $w \in L$ ist, ist $L \in \mathrm{P}$.