6.6 Die Klasse NP

Die TURING-Maschine M berechnet $f : L \longrightarrow \Sigma^*$ **nichtdeterministisch**, wenn es zu jedem $x \in L$ ein $y \in \Sigma^*$ gibt, so dass M mit der Verkettung xy von x und y als Input nach endlich vielen Schritten mit dem Output $f(x)$ anhält.

Beispiel. Sei $\Sigma = \mathbb{F}_2$ und $L = \{(n, a, x) \in \mathbb{N}^3 \mid n \geq 2, a, x \in \mathbb{M}_n\}$. Sei $f = {}^a\log \bmod n$ der diskrete Logarithmus.

Zu gegebenem x sei y der Logarithmus von x – woher wir ihn haben spielt in der Definition keine Rolle, er existiert jedenfalls. Dann muss die TURING-Maschine M nur noch prüfen, ob $a^y = x$, bevor sie y aufs Band schreiben und anhalten darf.

Vorstellung. Ein Kandidat y für die Lösung wird vorgegeben, M macht nur noch die Probe.

Alternativ-Vorstellung. Unbeschränkt viele *parallele* TURING-Maschinen probieren je ein $y \in \Sigma^*$ auf Eignung aus.

Die Menge **NP** („nichtdeteministisch-polynomiale Zeit“) ist definiert als die Menge der Funktionen, für die es eine TURING-Maschine M und eine natürliche Zahl $k \in \mathbb{N}$ gibt mit

(i) M berechnet f nichtdeterministisch,

(ii) $t_M(n) \leq n^k$ für fast alle $n \in \mathbb{N}$.

Es gelten die Inklusionen

$$\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{EXPTIME};$$

die erste davon ist trivial, die zweite ein Satz, der hier nicht bewiesen wird.

Das schon öfter angesprochene wichtigste ungelöste Problem der theoretischen Informatik ist die Vermutung

$$\mathbf{P} \neq \mathbf{NP}.$$

Ebenfalls unbewiesen ist die Vermutung

$$\mathbf{NP} \neq \mathbf{EXPTIME}.$$

Bewiesen ist allerdings

$$\mathbf{P} \neq \mathbf{EXPTIME},$$

wenn auch nur durch „künstliche“ Probleme; ein interessantes „natürliches“ Problem in der Differenzmenge ist nicht bekannt.

Die Kryptoanalyse schwieriger als **NP** zu machen, ist übrigens nicht möglich: Die Exhaustion – das Durchprobieren aller Schlüssel mit jeweiliger Probeverschlüsselung bei bekanntem Klartext – ist nämlich immer möglich und die Verschlüsselungsfunktion muss effizient, also in **P** sein.

Beispiele

1. Ist f der diskrete Logarithmus wie oben, so $f \in \mathbf{NP}$.
2. Genauso ist die Faktorisierung natürlicher Zahlen in **NP**.
3. Auch das Rucksackproblem ist in **NP**.

Die Funktion f heißt **NP-vollständig**, wenn es für jede TURING-Maschine M, die f berechnet (deterministisch!) und jede Funktion $g \in \mathbf{NP}$ eine TURING-Maschine N, die g berechnet, und eine natürliche Zahl $k \in \mathbb{N}$ gibt, so dass

$$t_N(n) \leq t_M(n)^k \quad \text{für fast alle } n \in \mathbb{N}.$$

D. h., die Komplexität von N ist höchstens polynomial in der Komplexität von M.

Vorstellung: **NP**-vollständige Probleme sind die maximal komplexen unter denen in **NP**.

Es gibt **NP***-vollständige Probleme.* – Das ist ein Satz, der hier nicht bewiesen wird. Z. B. ist das Rucksack-Problem **NP**-vollständig, ebenso die Nullstellenbestimmung von (Polynom-) Funktionen $p\colon \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$. Die Faktorisierung natürlicher Zahlen ist vermutlich nicht **NP**-vollständig.

Sollte $\mathbf{P} = \mathbf{NP}$ sein – was niemand glaubt –, so wären alle Funktionen in $\mathbf{P} = \mathbf{NP}$ auch **NP**-vollständig. Andernfalls gibt die folgende Skizze eine Vorstellung von der relativen Lage dieser Mengen: