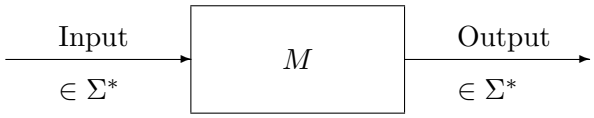
6.5 TURING-Maschinen

Die mathematische Komplexitätstheorie führt fast ausschließlich zu asymptotischen Aufwandsabschätzungen, und das sind so gut wie immer Abschätzungen nach oben. Sie beruht in ihren verschiedenen Varianten auf verschiedenen Berechnungs- oder Maschinen-Modellen. Hier wird die übliche Formalisierung von Komplexitätsaussagen durch TURING-Maschinen kurz skizziert.



Dabei ist Σ wie üblich ein endliches Alphabet. Der Input ist eine endliche Zeichenkette auf einem (in beide Richtungen unendlich langen) Band. Die TURING-Maschine M besitzt sie eine endliche Zustandsmenge, die unter anderem den Zustand „halt“ enthält. In Abhängigkeit vom Zustand führt sie gewisse Operationen aus, z. B. ein Zeichen vom Band lesen oder auf das Band schreiben und den Lesekopf um eine Stelle nach links oder rechts verschieben. Kommt M in den Zustand „halt“, so ist die dann auf dem Band befindliche Zeichenkette der Output.

Sei $L \subseteq \Sigma^*$ eine Sprache. Falls M für alle $x \in L$ nach endlich vielen Schritten den Zustand „halt“ erreicht, sagt man, M **akzeptiert die Sprache** L. Ist $f : L \longrightarrow \Sigma^*$ eine Funktion und kommt M für jedes $x \in L$ nach endlich vielen Schritten zum Zustand „halt“ mit dem Output $f(x)$, so sagt man, M **berechnet** f.

Mit etwas Mühe und nicht besonders elegant lassen sich alle Algorithmen im naiven Sinne durch TURING-Maschinen beschreiben. Die Komplexität lässt sich durch Zählen der Schritte ausdrücken; für den Input x braucht M bis zum Zustand „halt“ τ_x Schritte.

Meistens betrachtet man die „Worst-Case“-Komplexität. Sei wie üblich $L_n := L \cap \Sigma^n$. Dann wird die Funktion

$$t_M : \mathbb{N} \longrightarrow \mathbb{N}, \quad t_M(n) := \max\{\tau_x \mid x \in L_n\},$$

als **(Zeit-) Komplexität** der TURING-Maschine M (für L) bezeichnet.

Die Teilmenge **P** („polynomiale Zeit“) der Menge aller Funktionen aus Σ^* nach Σ^* wird so definiert: Die Funktion $f : L \longrightarrow \Sigma^*$ liegt in **P**, wenn es eine TURING-Maschine M und eine natürliche Zahl $k \in \mathbb{N}$ gibt mit

(i) M berechnet f,

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

Bemerkung. Äquivalent zu (ii) ist: Es gibt ein Polynom $p \in \mathbb{N}[X]$ mit $t_M(n) \leq p(n)$ für alle $n \in \mathbb{N}$.

Gibt es nämlich solch ein Polynom $p = a_r X^r + \cdots + a_0$ mit $a_r \neq 0$, so ist

$$\begin{aligned} a_r n^r &\geq a_{r-1} n^{r-1} + \cdots + a_0 \quad \text{für } n \geq n_0, \\ p(n) &\leq 2a_r n^r \quad \text{für } n \geq n_0, \\ p(n) &\leq n^{r+1} \quad \text{für } n \geq n_1 = \max\{2a_r, n_0\}. \end{aligned}$$

Ist umgekehrt $t_M(n) \leq n^k$ für $n \geq n_0$, so kann man $c \in \mathbb{N}$ wählen mit $t_M(n) \leq c$ für die endlich vielen $n = 0, \ldots, n_0 - 1$. Dann ist $t_M(n) \leq p(n)$ für alle $n \in \mathbb{N}$ mit $p = X^k + c$.

Analog ist die Menge **EXPTIME** („exponentielle Zeit“) definiert: f liegt in **EXPTIME**, wenn es eine TURING-Maschine M, eine natürliche Zahl $k \in \mathbb{N}$ und reelle Zahlen $a, b \in \mathbb{R}$ gibt mit

(i) M berechnet f,

(ii) $t_M(n) \leq a \cdot 2^{bn^k}$ für fast alle $n \in \mathbb{N}$.

Klar ist $\mathbf{P} \subseteq \mathbf{EXPTIME}$.

Beispiele mit $\Sigma = \mathbb{F}_2$.

1. Sei
$$L := \{(p, z) \in \mathbb{N}^2 \mid p \text{ prim } \equiv 3 \pmod 4,\ z \in \mathbb{M}_p^2\},$$
durch eine geeignete Binärdarstellung als Teilmenge von Σ^* codiert. Sei $f(p, z) =$ Quadratwurzel von $z \bmod p$ – ebenfalls als Element von Σ^* codiert. Dann ist $f \in \mathbf{P}$ nach 5.3.

2. Sei $L = \mathbb{N}_2$ die Menge aller natürlichen Zahlen ≥ 2 (binär codiert). Sei $f(x) =$ der kleinste Primfaktor von x. Dann ist $f \in \mathbf{EXPTIME}$ – man kann ja alle Zahlen $\leq \sqrt{x} \leq 2^{n/2}$ durchprobieren. *Vermutlich* ist aber $f \notin \mathbf{P}$.

3. **Das Rucksackproblem** (‘knapsack problem’). Hier ist
$$L = \{(m, a_1, \ldots, a_m, N) \mid m, a_1, \ldots, a_m, N \in \mathbb{N}\}$$
in geeigneter binärer Codierung,
$$f(m, a_1, \ldots, a_m, N) = \begin{cases} 1, & \text{wenn es } S \subseteq \{1, \ldots, m\} \text{ gibt} \\ & \text{mit } \sum_{i \in S} a_i = N, \\ 0 & \text{sonst.} \end{cases}$$
Dann ist $f \in \mathbf{EXPTIME}$ – man kann ja alle 2^m Teilmengen $S \subseteq \{1, \ldots, m\}$ durchprobieren. *Vermutlich* ist $f \notin \mathbf{P}$.