
17. Zeit- und Platzkomplexität von Turingmaschinen

Die Kosten der Rechnung einer Turingmaschine M bei Eingabe \vec{x} werden meist in der *Rechenzeit* $\text{time}_M(\vec{x})$ und dem *Speicherbedarf* $\text{space}_M(\vec{x})$ (s. Beispiel 16.2 im vorhergehenden Abschnitt) gemessen. Dabei ist es in der Komplexitätstheorie üblich, Mehrbandmaschinen zu betrachten, da diese i. Allg. schneller als 1-Band-Turingmaschinen sind, und man betrachtet Programme aus bedingten Anweisungen. Weiter beschränkt man sich nicht auf zahlentheoretische Funktionen, sondern betrachtet allgemeine Wortfunktionen, wobei wir uns jedoch auf den Fall 1-stelliger Funktionen $f : \Sigma^* \rightarrow \Sigma^*$ über einem Alphabet Σ beschränken werden. In der Regel betrachtet man die Kosten dann nicht für jedes einzelne Wort, sondern für alle Wörter gleicher Länge. Hierbei kann man die durchschnittlichen Kosten (*average case complexity*) oder die maximalen Kosten (*worst case complexity*) betrachten. Wir werden uns auf den zweiten Fall beschränken.

17.1 DEFINITION. Sei $f : \mathbb{N} \rightarrow \mathbb{N}$ eine total rekursive Funktion. Eine totale Mehrband-Turingmaschine M zur Berechnung einer 1-stelligen Funktion über dem Alphabet Σ ist *$f(n)$ -zeitbeschränkt*, falls

$$\forall x \in \Sigma^* (\text{time}_M(x) \leq f(|x|)),$$

und M ist *$f(n)$ -platzbeschränkt*, falls

$$\forall x \in \Sigma^* (\text{space}_M(x) \leq f(|x|)).$$

Hiermit können wir die Funktionen und Sprachen, die von einer Turingmaschine innerhalb einer gegebenen Zeit- bzw. Platzschränke berechnet werden, in Komplexitätsklassen zusammenfassen. Für eine total rekursive Zeitschranke $t : \mathbb{N} \rightarrow \mathbb{N}$ definieren wir

$$\begin{aligned} \text{FDTIME}_k^m(t(n)) &= \{f : \Sigma_m^* \rightarrow \Sigma_m^* \text{ total : } f \text{ wird von einer } \\ &\quad t(n)\text{-zeitbeschränkten } k\text{-Band-TM berechnet}\} \\ \text{FDTIME}^m(t(n)) &= \{f : \Sigma_m^* \rightarrow \Sigma_m^* \text{ total : } f \text{ wird von einer } \\ &\quad t(n)\text{-zeitbeschränkten Mehrband-TM berechnet}\} \\ \text{DTIME}_k^m(t(n)) &= \{A \subseteq \Sigma^* : c_A \text{ wird von einer } t(n)\text{-zeitbeschränkten } \\ &\quad k\text{-Band-TM berechnet}\} \\ \text{DTIME}^m(t(n)) &= \{A \subseteq \Sigma^* : c_A \text{ wird von einer } t(n)\text{-zeitbeschränkten } \\ &\quad \text{Mehrband-TM berechnet}\} \end{aligned}$$

Für eine total rekursive Platzschränke $s : \mathbb{N} \rightarrow \mathbb{N}$ definiert man entsprechend die Platzklassen

$$\begin{aligned} &\text{FDSpace}_k^m(s(n)), \text{FDSpace}^m(s(n)), \\ &\text{DSpace}_k^m(s(n)), \text{DSpace}^m(s(n)), \end{aligned}$$

indem man $s(n)$ -platzbeschränkte Turingmaschinen betrachtet.

Im folgenden werden wir meist das binäre Alphabet Σ_2 betrachten. In diesem Fall lassen wir den Index $m = 2$ bei obigen Komplexitätsklassen fort. Weiter werden wir uns auf den Vergleich der Sprachkomplexitätsklassen beschränken.

BEMERKUNG. Betrachtet man Probleme (oder Funktionen) über anderen Daten als Wörtern, so muss man diese durch geeignete Kodierung durch Wortprobleme darstellen. Bei arithmetischen Problemen benutzt man dabei in der Komplexitätstheorie jedoch (anders als in der Rekursionstheorie) in der Regel die Binär- und nicht die Unärarstellung, da letztere unverhältnismäßig lange Darstellungen liefert. Unter einer (oberen) Schranke des ursprünglichen Problems versteht man dann eine (obere) Schranke des zugehörigen Wortproblems.

Bei Zeitschranken geht man davon aus, dass $t(n) \geq n + 1$ gilt, und liest daher $t(n)$ als $\max(t(n), n + 1)$. (Zeitschranken t mit $t(n_0) \leq n_0$ für ein n_0 erlauben für Eingaben der Länge $> n_0$ nicht, diese vollständig zu lesen, und führen zu konstanten Zeitschranken (s. Übung).) Bei Platzschranken betrachtet man auch sublineare Schranken. Hierzu muss man jedoch off-line Turingmaschinen betrachten und dann beim Platzbedarf das Eingabeband nicht mitrechnen. Hier kann man dann bis $s(n) = \log(n)^1$ heruntergehen, während kleinere Schranken wieder auf konstante Schranken kollabieren. (Der Grund hierfür ist, dass ein Binärzähler von 0 bis n mit $\log(n)$ Platz auskommt.)

Ist $C(f(n))$ eine Komplexitätsklasse wie oben, und ist $F = \{f_k(n) : k \geq 1\}$ eine Familie total rekursiver Funktionen, so schreiben wir

$$C(F) = \bigcup_{f \in F} C(f(n)).$$

Insbesondere steht $C(O(f))$ für $\bigcup_{k \geq 1} C(kf(n) + k)$. Wichtige Zeitkomplexitätsklassen sind durch solche natürlich gewählten Familien bestimmt:

$$\begin{aligned} \text{LIN} &:= \text{DTIME}(O(n)) && (\text{Det. Linearzeit}) \\ \text{P} &:= \text{PTIME} && (\text{Det. Polynomialzeit}) \\ &:= \text{DTIME}(\text{poly}(n)) = \bigcup_p \text{Polynom} \text{DTIME}(p(n)) \\ \text{E} &:= \text{DTIME}(2^{O(n)}) && (\text{Det. lineare Exponentialzeit}) \\ \text{EXP} &:= \text{DTIME}(2^{\text{poly}(n)}) && (\text{Det. polyn. Exponentialzeit}) \\ &:= \bigcup_p \text{Polynom} \text{DTIME}(2^{p(n)}) \\ \text{ELEMENTARY} &:= \bigcup_{k \geq 1} \text{DTIME}(g^k(O(n))) \\ &&& (\text{mit } g(n) = 2^n \text{ und } g^k = \underbrace{g \circ \dots \circ g}_{k\text{-mal}}) \end{aligned}$$

Weiter lassen sich primitiv rekursive und rekursive Wortmengen definieren durch

$$\begin{aligned} \text{PRIM} &:= \bigcup_{t \text{ primitiv rekursiv}} \text{DTIME}(t(n)) && (\text{primitiv rekursiv}) \\ \text{REK} &:= \bigcup_{t \text{ total rekursiv}} \text{DTIME}(t(n)) \end{aligned}$$

(Für Zahlenmengen stimmt dies mit der ursprünglichen Definition überein.)

Zu den wichtigsten Platzklassen gehören:

$$\begin{aligned} \text{LOGSPACE} &:= \text{DSPACE}(\log(O(n))) && (\text{logarithmisch Platz}) \\ \text{PSPACE} &:= \text{DSPACE}(\text{poly}(n)) && (\text{polynomiell Platz}) \\ \text{EXSPACE} &:= \text{DSPACE}(2^{\text{poly}(n)}) && (\text{polynomiell Exponentialplatz}) \end{aligned}$$

¹Reellwertige Schranken wie $\log(n)$ werden stets auf die nächste natürliche Zahl aufgerundet.

Im folgenden werden wir auf die Beziehung zwischen diesen Komplexitätsklassen und den elementaren Komplexitätsklassen im Allgemeinen eingehen. Zwischen den elementaren Komplexitätsklassen bestehen aufgrund der Definition folgende triviale Beziehungen:

$$\begin{aligned} \text{Für } k \leq k' \text{ gilt } \text{DTIME}_k(f(n)) \subseteq \text{DTIME}_{k'}(f(n)) \subseteq \text{DTIME}(f(n)) \text{ und} \\ \text{DSPACE}_k(f(n)) \subseteq \text{DSPACE}_{k'}(f(n)) \subseteq \text{DSPACE}(f(n)) \quad (17.1) \end{aligned}$$

$$\begin{aligned} \text{DTIME}(f(n)) = \bigcup_{k \geq 1} \text{DTIME}_k(f(n)) \text{ und} \\ \text{DSPACE}(f(n)) = \bigcup_{k \geq 1} \text{DSPACE}_k(f(n)) \quad (17.2) \end{aligned}$$

$$\begin{aligned} \text{Gilt } f(n) \leq f'(n) \text{ fast überall, so gilt } \text{DTIME}_{(k)}(f(n)) \subseteq \text{DTIME}_{(k)}(f'(n)) \text{ und} \\ \text{DSPACE}_{(k)}(f(n)) \subseteq \text{DSPACE}_{(k)}(f'(n)) \quad (17.3) \end{aligned}$$

Der in Abschnitt 6 bewiesene Bandreduktionssatz zusammen mit den dort bereits beobachteten Komplexitätsschranken erlaubt uns die Komplexität von Problemen bzgl. 1-Band-Turingmaschinen durch deren Komplexität bzgl. Mehrband-Turingmaschinen abzuschätzen:

17.2 SATZ. (1. BANDREDUKTIONSSATZ) *Zu jeder $t(n)$ -zeit- und $s(n)$ -platzbeschränkten Mehrband-Turingmaschine gibt es eine äquivalente $O(t(n)^2)$ -zeit- und $O(s(n))$ -platzbeschränkte 1-Band-Turingmaschine.*

17.3 KOROLLAR. $\text{DSPACE}(s(n)) \subseteq \text{DSPACE}_1(O(s(n)))$ und $\text{DTIME}(t(n)) \subseteq \text{DTIME}_1(O(t(n)^2))$. \square

Führt die Reduktion auf ein Band für die Platzkomplexität auf einen nur geringen (wie wir unten sehen werden – vernachlässigbaren) linearen Verlust, so ist der Verlust bei der Zeit quadratisch, also sehr beachtlich. Dieser Zeitverlust lässt sich i. Allg. nicht vermeiden. Für die Sprache $L = \{ww^R : w \in \{0, 1\}^*\}$, wobei w^R das Spiegelwort von w ist, gilt $L \in \text{DTIME}_2(O(n))$, aber $L \notin \text{DTIME}(t(n))$ für jede subquadratische Zeitschranke t (modulo eines linearen Faktors), d.h. t mit

$$\liminf_{n \rightarrow \infty} \frac{t(n)}{n^2} = 0.$$

Die Reduktion auf zwei Bänder lässt sich jedoch mit weit geringerem Zeitverlust bewerkstelligen:

17.4 SATZ. (2. BANDREDUKTIONSSATZ) *Zu jeder $t(n)$ -zeitbeschränkten Mehrband-Turingmaschine M gibt es eine äquivalente $O(t(n) \log(t(n)))$ -zeitbeschränkte 2-Band-Turingmaschine M' . Insbesondere gilt also*

$$\text{DTIME}(t(n)) \subseteq \text{DTIME}_2(O(t(n) \log(t(n))).$$

BEWEIS. Siehe Literatur oder Vorlesung „Komplexitätstheorie“.

Wie schon oben erwähnt, sind lineare Faktoren bei den Turingmaschinen-Komplexitätsklassen DTIME und DSPACE bedeutungslos.

17.5 SATZ. (LINEARE KOMPRESSION) Sei $s(n)$ eine hyperlogarithmische (d.h. $s(n) \geq c \log(n)$ f.ü. für alle $c \geq 1$) rekursive Funktion, und sei $c \geq 1$. Zu jeder $s(n)$ -platzbeschränkten k -Band-Turingmaschine gibt es eine äquivalente $s(n)/c$ -platzbeschränkte k -Band-Turingmaschine M' .

BEWEISIDEE. Die Maschine M' simuliert die Maschine M Schritt für Schritt, wobei sie c benachbarte Felder (auf einem Arbeitsband) zu einem Feld zusammenfasst, d.h. für die Bandalphabet Γ von M und Γ' von M' gilt $\Gamma^c \subseteq \Gamma'$. In ihrem Zustand merkt sich M' neben dem M -Zustand noch die Komponente $i \in \{1, \dots, c\}$ des Arbeitsfeldes von M' , die M -Arbeitsfeld ist. \square

Mit derselben Grundidee erhält man auch (für $k \geq 2$) eine lineare Beschleunigung. (Details in der Vorlesung „Komplexitätstheorie“.)

17.6 SATZ. (LINEARE BESCHLEUNIGUNG) Sei $t(n)$ eine hyperlineare, rekursive Funktion, und sei $c \geq 1$. Zu jeder $t(n)$ -zeitbeschränkten k -Band-Turingmaschine M ($k \geq 2$) gibt es eine äquivalente $t(n)/c$ -zeitbeschränkte k -Band-Turingmaschine M' . Insbesondere gilt also $\text{DTIME}_{(k)}(t(n)) = \text{DTIME}_{(k)}(O(t(n)))$.

Die Sätze 17.5 und 17.6 zeigen, dass bei den Kostenmaßen für Turingmaschinen lineare Lücken auftreten. Dieses, auf den ersten Blick der Intuition widersprechende Phänomen ergibt sich daraus, dass wir eine ganze Familie von Basismaschinen zulassen, die auf einem Feld unterschiedlich viel Informationen abspeichern können, sich also im Leistungsumfang eines Einzelschrittes unterscheiden. Würden wir das Bandalphabet (und die Anzahl der Bänder) festhalten (oder die Kosten mit der Größe des Bandalphabetes multiplizieren), so würden diese linearen Lücken i.a. nicht mehr auftreten.

Dual zu der linearen Kompression und Beschleunigung führt eine Normierung des Bandalphabetes zu einer linearen Kostenerhöhung, wie wir in Abschnitt 11 (s. Lemma 11.3) bereits gesehen haben. Die dort durchgeführten Überlegungen für 1-Band-Turingmaschinen lassen sich auf k -Band-Turingmaschinen für jedes $k \geq 1$ übertragen, sodass sich folgende Ergebnisse zur Normierbarkeit und zur Existenz universeller Turingmaschinen ergeben:

17.7 SATZ. (NORMIERBARKEIT) Zu jeder $t(n)$ -zeit- und $s(n)$ -platzbeschränkten k -Band-Turingmaschine M ($k \geq 1$) zur Erkennung einer Sprache $A \subseteq \Sigma_2^*$ (d.h. zur Berechnung von c_A) gibt es eine äquivalente normierte k -Band-Turingmaschine M' , die $O(t(n))$ -zeit- und $O(s(n))$ -platzbeschränkt ist.

17.8 SATZ. (EXISTENZ UNIVERSELLER TURINGMASCHINEN) Für jedes $k \geq 1$ gibt es eine Mehrband-Turingmaschine U_k mit folgenden Eigenschaften. Ist M eine $t(n)$ -zeit- und $s(n)$ -platzbeschränkte normierte k -Band-Turingmaschine zur Erkennung einer Sprache $A \subseteq \Sigma_2^*$, so gibt es ein Wort $w \in \Sigma_2^*$ mit

$$(i) \quad \forall x \in \Sigma_2^* (\Phi_M(x) = \Phi_{U_k}(w, x))$$

$$(ii) \quad \text{time}_{U_k}(w, x) \in O(t(|x|)) \ \& \ \text{space}_{U_k}(w, x) \in O(s(|x|))$$

Kombiniert man die Bandreduktionssätze mit Normierung und Existenz universeller Maschinen, so kann man die Zeit- und Platzkomplexitätsklassen durch Zeit- und Platzbedarf der universellen Maschinen U_1 und U_2 wie folgt abschätzen:

17.9 KOROLLAR. (a) Die Klasse $\text{DSPACE}(s(n))$ ist enthalten in

$$\{A \subseteq \Sigma_2^* : \exists w(\forall x(\varphi_{U_1}(w, x) = c_A(x)) \ \& \ \text{space}_{U_1}(w, x) \in O(s(|x|)))\}$$

(b) Die Klasse $\text{DTIME}(t(n))$ ist enthalten in

$$\{A \subseteq \Sigma_2^* : \exists w(\forall x(\varphi_{U_2}(w, x) = c_A(x)) \ \& \ \text{time}_{U_2}(w, x) \in O(t(|x|) \log(t(|x|))))\}$$

BEWEIS. Wir zeigen Teil (a) des Korollars. Es gilt:

$$\begin{aligned} \text{DSPACE}(s(n)) &\subseteq \text{DSPACE}_1(O(s(n))) && \text{(1. Bandred.satz)} \\ &\subseteq \{A \in \Sigma_2^* : \text{Es gibt eine normierte, } O(s(n))\text{-} \\ &\quad \text{platzbeschränkte 1-Band-TM,} \\ &\quad \text{die } c_A \text{ berechnet}\} && \text{(Normierung)} \\ &\subseteq \{A \in \Sigma_2^* : \exists w \text{ mit Eigenschaften (i) und (ii)} \\ &\quad \text{in Satz 17.8}\} && \text{(Satz über univ. TM)} \end{aligned}$$

□

Wir werden diese Abschätzungen zum Beweis der Hierarchiesätze im nächsten Abschnitt benutzen.

Wir beschliessen diesen Abschnitt mit dem Vergleich von Zeit- und Platzkomplexität.

17.10 SATZ. (a) Jede $t(n)$ -zeitbeschränkte Turingmaschine ist auch $t(n)$ -platzbeschränkt.

(b) Jede $s(n)$ -platzbeschränkte Turingmaschine ist $O(2^{O(s(n))})$ -zeitbeschränkt.

17.11 KOROLLAR. (i) $\text{DTIME}_{(k)}(t(n)) \subseteq \text{DSPACE}_{(k)}(t(n))$

(ii) $\text{DSPACE}_{(k)}(s(n)) \subseteq \text{DTIME}_{(k)}(2^{O(s(n))})$

BEWEIS VON SATZ 17.10. Zum Beweis von Teil (a) genügt es zu beobachten, dass die Laufzeit $t(n)$ einer Turingmaschine zugleich die Anzahl der neu besuchten Felder auf jedem Band beschränkt. Also ist $t(n) + 1$ und damit $t(n)$ (wg. linearer Kompression) eine obere Platzschränke.

Zum Beweis von (b) sei $M = (B, P)$ eine $s(n)$ -platzbeschränkte k -Band-Turingmaschine, und sei x ein Wort der Länge n über dem Eingabealphabet Σ . Da M für jede Eingabe terminiert, wiederholt sich keine Konfiguration in der Rechnung von M bei Eingabe x , weshalb

$$\text{time}_M(x) \leq \text{kon}_M(x),$$

wobei $\text{kon}_M(x)$ die Kardinalität der Menge der theoretisch möglichen Konfigurationen bei Eingabe x ist, d.h. die Menge der Konfigurationen mit x auf dem Eingabeband (nur relevant im Falle, dass s sublinear und M daher eine off-line Turingmaschine ist) und relevanten Bandinschriften auf den Arbeitsbändern innerhalb des Adressintervalls $(-s(n), s(n))$. Diese kann man aber abschätzen durch

$$\text{kon}_M(x) \leq (n+2)(\gamma^{2s(n)})^k (2s(n))^k z,$$

wobei γ die Kardinalität des Bandalphabetes und z die Anzahl der Zustände ist. (Dann ist nämlich $(n+2)$ die Anzahl der möglichen Positionen des Arbeitsfeldes auf dem Eingabeband, $\gamma^{2s(n)}$ die Anzahl der möglichen Inschriften eines jeden der k Arbeitsbänder und $2s(n)$ die Anzahl der möglichen Positionen des Arbeitsfeldes auf diesen.) Da nach Annahme $s(n) \geq \log(n)$, also $n \leq 2^{s(n)}$ gilt, ergeben sich hieraus Konstanten c, c' mit

$$\text{kon}_M(x) \leq c^{s(|x|)} \leq 2^{c's(|x|)}$$

für fast alle x . □

BEMERKUNG. Für „natürliche“ Schranken t lässt sich die Abschätzung (a) in Korollar 17.11 geringfügig verbessern. Die Verbesserung ist aber so marginal, dass sie für die Praxis bedeutungslos ist. Ob sich die Abschätzung (b) verbessern lässt, ist unbekannt.