
18. Platz- und Zeithierarchiesätze

Wir untersuchen hier die Frage, wie man zu einer gegebenen Platzschranke $s(n)$ eine neue Schranke $S(n)$ finden kann, die die Lösung zusätzlicher Probleme ermöglicht (und entsprechend für Zeitschranken). Notwendige Anforderungen an solch eine Schranke ergeben sich aus der Linearen Kompression und dem allgemeinen Lückensatz. Aus Ersterem ergibt sich die Notwendigkeit, dass $S(n) \notin O(s(n))$ gelten muss. Aus Letzterem folgt, dass es nicht genügt, $s(n)$ um einen vorgegebenen Wachstumsfaktor zu $S(n)$ zu vergrößern: Um die Existenz einer Sprache $A \in \text{DSPACE}(S(n)) - \text{DSPACE}(s(n))$ zu sichern, müssen wir explizit fordern, dass es eine Platzkostenfunktion $\sigma(n)$ zwischen $s(n)$ und $S(n)$ gibt (wobei zusätzlich – wegen der linearen Kompression – $\sigma(n) \notin O(s(n))$ gelten muss). Wie wir sehen werden, sind diese Forderungen im wesentlichen auch hinreichend. Für die hier von uns betrachtete worst-case-Komplexität kann man die Definition von Platzkostenfunktionen wie folgt approximieren.

18.1 DEFINITION. Eine rekursive Funktion $s(n) \geq \log(n)$ ist *platzkonstruierbar*, wenn es eine Mehrband-Turingmaschine M gibt, für die $\text{space}_M(x) = s(|x|)$ für alle Wörter x gilt.

Die üblichen monotonen Funktionen, die man in der Arithmetik betrachtet, wie $\log(n)$, alle Polynome, 2^n , $n!$ etc. sind alle platzkonstruierbar, und jede rekursive Funktion wird von einer platzkonstruierbaren Funktion majorisiert (siehe Übungen). Die Beschränkung auf platzkonstruierbare Platzschranken ist für praktische Überlegungen daher unwesentlich.

18.2 SATZ. (PLATZHIERARCHIESATZ) Seien $s(n), S(n) \geq \log(n)$ rekursive Funktionen, für die

(S1) $S(n) \notin O(s(n))$ und

(S2) $S(n)$ platzkonstruierbar

gilt. Dann gilt $\text{DSPACE}(S(n)) \not\subseteq \text{DSPACE}(s(n))$. Werden $s(n)$ und $S(n)$ zusätzlich so gewählt, dass $s(n) \leq S(n)$ für fast alle n gilt, so gilt also

$$\text{DSPACE}(s(n)) \subsetneq \text{DSPACE}(S(n)).$$

BEWEIS. Wir geben eine $O(S(n))$ -platzbeschränkte Mehrband-Turingmaschine M an, so dass die von M erkannte Sprache $L = L_M$ nicht in $\text{DSPACE}(s(n))$ liegt. Wegen der linearen Kompression gilt dann

$$L \in \text{DSPACE}(S(n)) - \text{DSPACE}(s(n)).$$

Hierbei wird $L \notin \text{DSPACE}(s(n))$ durch Diagonalisierung mit Hilfe der universellen 1-Band-Turingmaschine U_1 erreicht.

Die $O(S(n))$ -Platzbeschränktheit wird wie folgt sichergestellt. Bei jeder Eingabe x , $|x| = n$, werden zunächst auf den ersten fünf Arbeitsbändern die Felder mit Adressen im Intervall $(-S(n), S(n))$ markiert. Da $S(n)$ platzkonstruierbar ist, kann dies mit Hilfe einer Turingmaschine mit Platzbedarf $S(n)$ (die eventuell auf weitere Bänder zugreift) realisiert werden. Die Bänder 1-3 werden dann zur Simulation von U_1 verwendet. Auf Band 4 wird ein Binärzähler der Länge $2S(n)$ initialisiert, der dann in jedem Simulationsschritt inkrementiert wird. Verlässt U_1 den markierten Bereich während der Simulation, oder läuft der Zähler über, so wird die Simulation sofort beendet und die Eingabe x verworfen.

Die Simulation von U_1 durch M sieht nun wie folgt aus. M zerlegt die Eingabe x in $x = 1^{|w|}0wv$ und schreibt w auf das 5. Arbeitsband, (falls hierzu der markierte Platz ausreicht). M simuliert dann U_1 für Eingabe (w, x) , wobei die zugehörige Inschrift wBx des Eingabebandes von U_1 bei M auf das 5. Band (w) und das Eingabeband (x) aufgeteilt ist. Kann die Simulation innerhalb des bereitgestellten Platzes und der erlaubten Zeit beendet werden, so akzeptiert M die Eingabe x genau dann, wenn U_1 die Eingabe (w, x) verwirft.

Dass auf diese Art gegen alle Sprachen in $DSPACE(s(n))$ diagonalisiert wird, ergibt sich aus der Wahl von $S(n)$ und der Tatsache, dass die universelle Maschine U_1 mit linear beschränktem Platzverlust auskommt. Zum formalen Nachweis gehen wir indirekt von der Widerspruchsannahme, dass $L \in DSPACE(s(n))$ gilt, aus. Nach Korollar 17.9 gibt es dann ein Wort w , so dass

$$\forall x (c_L(x) = \varphi_{U_1}(w, x)) \quad (18.1)$$

und

$$\text{space}_{U_1}(w, x) \in O(s(x)).$$

Wie im Beweis von Korollar 17.11(ii) ergibt sich hieraus weiter, dass

$$\text{time}_{U_1}(w, x) \in O(2^{O(s(|x|))}).$$

Wegen (S1) gibt es daher unendlich viele Zahlen n mit

$$\forall x (|x| = n \Rightarrow \text{space}_{U_1}(w, x) < S(n) \ \& \ \text{time}_{U_1}(w, x) < 2^{S(n)}). \quad (18.2)$$

Für ein n mit $n > 2|w| + 1$ und $S(n) > |w|$, das (??) erfüllt, und für $x = 1^{|w|}0w0^{n-(2|w|+1)}$ wird daher von M bei Eingabe x die Simulation von U_1 für Eingabe (w, x) abgeschlossen, also

$$c_L(x) = c_{L_M}(x) = 1 - \varphi_{U_1}(w, x)$$

sichergestellt. Dies aber widerspricht (??). □

Aus dem Platzhierarchiesatz ergeben sich z.B. folgende echte Inklusionen:

$$\begin{array}{lcl} DSPACE(\log(n)) & \subset & DSPACE(n) \\ DSPACE(n^k) & \subset & DSPACE(n^k \log(n)) \subset DSPACE(n^{k+1}) \\ DSPACE(n^k) & \subset & DSPACE(n^{\log(n)}) \end{array}$$

Für die Zeitkomplexität erhält man folgenden Hierarchiesatz (Beweis in der Vorlesung „Komplexitätstheorie“).

18.3 SATZ. (ZEITHIERARCHIESATZ) Seien $t(n), T(n) > n$ rekursive Funktionen, für die

(T1) $T(n) \notin O(t(n) \log(t(n)))$ und

(T2) $T(n)$ zeitkonstruierbar

gilt. Dann gilt $\text{DTIME}(T(n)) \not\subseteq \text{DTIME}(t(n))$. Sind t und T so gewählt, dass zusätzlich $t(n) \leq T(n)$ fast überall gilt, so gilt also

$$\text{DTIME}(t(n)) \subsetneq \text{DTIME}(T(n)).$$

Hierbei ist eine rekursive Funktion $f(n)$ zeitkonstruierbar, wenn es eine Mehrband-Turingmaschine M gibt, deren Rechenzeit bei jeder Eingabe der Länge n gerade $f(n)$ beträgt.

Den Zeithierarchiesatz beweist man analog zum Plathierarchiesatz, wobei man jedoch die universelle Maschine U_2 (statt U_1) simuliert. Parallel zur Simulation lässt man eine Turingmaschine mit exaktem Zeitbedarf $T(n)$ laufen, um so $L = L_M \in \text{DTIME}(T(n))$ zu sichern. Den Erfolg der Diagonalisierung weist man mit Hilfe von Korollar 17.9 (b) nach. Der zusätzliche Faktor $\log(t(n))$ in (T1) (und in Korollar 17.9 (b)) ergibt sich aus dem Zeitverlust bei der Reduktion auf zwei Bänder (vgl. Satz 17.4). Da unbekannt ist, ob die Reduktion auf zwei (oder ein anderes festes k) Bänder mit geringerem Zeitverlust möglich ist, ist ebenfalls unbekannt, ob man die Voraussetzung (T1) im Zeithierarchiesatz abschwächen kann.