Bemerkungen:

- Ist der Wert von $S(n)$ (und damit auch τ) unbekannt, so führt man das Verfahren nacheinander mit den Werten

$$\log n, 2\log n, \ldots$$

  als Platzschranke aus. Dabei überprüft man für jeden dieser Werte, ob er zu klein ist, d.h., ob die NDTM eine Konfiguration innerhalb des gegebenen Platzes erreicht, bei der aber eine Nachfolgekonfiguration über die Platzschranke hinaus führt. Nur wenn dies der Fall ist (das kann leicht mit Hilfe der Prozedur reach überprüft werden), wird die Platzschranke weiter hochgesetzt.
- Ob auch NSPACE$(S(n))$ für $S(n) = o(\log n)$ unter Komplement abgeschlossen ist, ist ein offenes Problem (die Vermutung ist, dass eher nicht).

6. Hierarchiesätze

Satz 189

Sei $S(n)$ (bzw. $T(n)$) eine berechenbare totale Funktion. Dann gibt es eine rekursive Sprache L mit

$$L \notin \mathit{DSPACE}(S(n)) \text{ (bzw. } \mathit{DTIME}(T(n)))$$

Beweis:

Der Beweis erfolgt durch Diagonalisierung. Sei $w_0, w_1, w_2, \ldots$ eine rekursive Auflistung von $\{0,1\}^*$, $M_0, M_1, M_2, \ldots$ eine Gödelisierung der (deterministischen) TM mit Eingabealphabet $\{0,1\}^*$.

Beweis (Forts.):

Betrachte die Sprache

$$L = \{w_i;\ M_i \text{ akzeptiert } w_i \text{ nicht innerhalb Zeit } T(|w_i|)\}.$$

L ist rekursiv. Annahme, $L = L(M_i)$ für eine $T(n)$-zeitbeschränkte (det.) TM M_i.

Falls $w_i \in L$, gilt per Definition, dass M_i w_i nicht in Zeit $T(|w_i|)$ akzeptiert, also Widerspruch.

Falls $w_i \notin L = L(M_i)$, akzeptiert M_i w_i nicht, also müsste per Definition $w_i \in L$ sein, also wiederum Widerspruch! □

6.1 Eine Platzhierarchie

Lemma 190

Sei L eine Sprache, die von einer $S(n)$-platzbeschränkten (det.) TM akzeptiert wird ($S(n) \geq \log n$). Dann gibt es eine $S(n)$-platzbeschränkte (det.) TM für L, die auf allen Eingaben hält (also L erkennt).

Beweis:

Man führe einen Zähler mit, der die Laufzeit in Abhängigkeit von der maximalen Anzahl von bisher besuchten Arbeitsbandfeldern beschränkt. □

Definition 191

Eine Funktion $S(n)$ heißt platzkonstruierbar, falls es eine $S(n)$-platzbeschränkte DTM gibt, die bei Eingabe $w \in \{0,1\}^n$ $S(n)$ in unärer Darstellung berechnet.

Satz 192 (Deterministischer Platzhierarchiesatz)

Sei $S_2(n) \geq \log n$ und platzkonstruierbar, sei $S_1(n) = o(S_2(n))$. Dann ist

$$\mathit{DSPACE}(S_1(n)) \subset \mathit{DSPACE}(S_2(n))\,.$$

Beweis:

Wir konstruieren eine DTM M, die bei Eingabe w zunächst in Platz $S_2(|w|)$ $S_2(|w|)$ Arbeitsbandfelder absteckt (und in der weiteren Rechnung nie mehr Platz benützt). M simuliert dann M_w auf Eingabe w und akzeptiert w genau dann, falls M die Simulation von M_w auf w in der vorgegebenen Platzschranke durchführen kann und M_w dabei w nicht akzeptiert.

Offensichtlich ist $L(M) \in \mathrm{DSPACE}(S_2(n))$. Die Annahme $L(M) \in \mathrm{DSPACE}(S_1(n))$ führt wie oben zu einem Widerspruch. $\square$

6.2 Eine Zeithierarchie

Definition 193

Eine Funktion $T(n)$ heißt zeitkonstruierbar, falls es eine DTM gibt, die bei Eingabe $w \in \{0,1\}^n$ nach genau $T(n)$ Schritten hält.

Ein Ergebnis wie soeben für DSPACE kann man für DTIME nicht zeigen, da für eine „real-time"-Simulation einer k-Band-TM wieder k Bänder benötigt werden, die simulierende TM aber eine fest vorgegebene Anzahl von Arbeitsbändern hat. Man kann aber zeigen:

Satz 194

$$\mathit{DTIME}(T(n)) \subseteq \mathit{DTIME}_2(T(n)\log T(n))\,,$$

wobei $\mathit{DTIME}_2(T(n))$ *die auf 2-Band-DTMs in Zeit* $T(n)$ *erkennbaren Sprachen bezeichnet.*

Satz 195 (Allgemeiner deterministischer Zeithierarchiesatz)

Sei $T_2(n)$ *zeitkonstruierbar und sei* $T_1(n)\log T_1(n) = o(T_2(n))$. *Dann ist*

$$\mathit{DTIME}(T_1(n)) \subset \mathit{DTIME}(T_2(n))\,.$$

Beweis:

Analog zum Platzhierarchiesatz. $\square$

Man kann auch zeigen

Satz 196

Sei $k \geq 2$, $T_2(n)$ *zeitkonstruierbar und* $T_1(n) = o(T_2(n))$. *Dann ist*

$$\textit{DTIME}_k(T_1(n)) \subset \textit{DTIME}_k(T_2(n))\,.$$

Beweis:

[Idee] Der Beweis beruht auf einer Methode, bei der Simulation einer k-Band-DTM auf den gleichen k Bändern eine „Uhr“ für $T_2(n)$ mitlaufen zu lassen, ohne dafür einen Zeitverlust von mehr als einem konstanten Faktor in Kauf nehmen zu müssen. Dies gelingt mit Hilfe eines so genannten distributiven Zählers, bei dem die Darstellung der verbleibenden Zeit (also des Zählers) geschickt über das ganze Arbeitsband verteilt wird.

Beweis (Forts.):

Details dazu findet man z.B. im Buch:

Karl Rüdiger Reischuk:
Komplexitätstheorie — Band I: Grundlagen.
B.G. Teubner, Stuttgart-Leipzig, 1999

□