

Definition 178

Sei $T(n)$ eine Zeitschranke, $S(n)$ eine Platzschranke. Wir definieren die folgenden Komplexitätsklassen:

- 1 $\text{DTIME}(T(n))$ ist die Klasse aller Probleme, die von einer deterministischen Turingmaschine in Zeit $T(n)$ erkannt werden können.
- 2 $\text{NTIME}(T(n))$ ist die Klasse aller Probleme, die von einer nichtdeterministischen Turingmaschine in Zeit $T(n)$ akzeptiert werden können.
- 3 $\text{DSPACE}(S(n))$ ist die Klasse aller Probleme, die von einer deterministischen Turingmaschine in Platz $S(n)$ erkannt werden können.
- 4 $\text{NSPACE}(S(n))$ ist die Klasse aller Probleme, die von einer nichtdeterministischen Turingmaschine in Platz $S(n)$ akzeptiert werden können.

2. Linearer Speed-up, lineare Bandkompression, Bandreduktion

Satz 179

Falls L von einer $S(n)$ -platzbeschränkten k -Band TM (DTM oder NDTM) erkannt bzw. akzeptiert wird, so auch von einer $k \cdot S(n)$ -platzbeschränkten 1-Band-TM.

Beweis:

Einfache Simulation!

Die folgenden Behauptungen ergeben sich durch einfache Konstruktionen unter Verwendung eines geeignet größeren Bandalphabets. Details bleiben als Übungsaufgabe überlassen.

Satz 180 (Bandkompression)

Falls L von einer $S(n)$ -platzbeschränkten k -Band-TM akzeptiert/erkannt wird, dann auch, für jedes $c > 0$, von einer $c \cdot S(n)$ -platzbeschränkten k -Band-TM.

Korollar 181

Sei $L \in \text{NSPACE}(S(n))$, $c > 0$, dann ist auch $L \in \text{NSPACE}(c \cdot S(n))$.

Satz 182 (Linearer Speed-up)

Falls L von einer $T(n)$ -zeitbeschränkten k -Band-TM akzeptiert/erkannt wird, dann auch, für jedes $c > 0$, von einer $c \cdot T(n)$ -zeitbeschränkten k -Band-TM, vorausgesetzt, dass $k \geq 2$ und $T(n) = \omega(n)$.

Korollar 183

Falls $T(n) = \omega(n)$ und $c > 0$, dann gilt

$$DTIME(T(n)) = DTIME(cT(n)).$$

Satz 184

Falls L von einer cn -zeitbeschränkten k -Band-TM akzeptiert/erkannt wird ($c \geq 1$), dann auch, für jedes $\epsilon > 0$, von einer $(1 + \epsilon)n$ -zeitbeschränkten k -Band-TM, vorausgesetzt, dass $k \geq 2$.

Definition 185 (wichtige Komplexitätsklassen)

1

$$\mathcal{P} = \bigcup_{c>0, k>0} \text{DTIME}(cn^k);$$

2

$$\mathcal{NP} = \bigcup_{c>0, k>0} \text{NTIME}(cn^k);$$

3

$$\mathcal{L} = \bigcup_{c>0} \text{DSpace}(c \log n);$$

4

$$\mathcal{NL} = \bigcup_{c>0} \text{NSpace}(c \log n);$$

Definition 185 (wichtige Komplexitätsklassen)

5

$$\text{PSPACE} = \bigcup_{c>0, k>0} \text{DSPACE}(cn^k);$$

6

$$\text{NPSPACE} = \bigcup_{c>0, k>0} \text{NSPACE}(cn^k);$$

3. Zeit und Platz

Es ist klar, dass eine $T(n)$ zeitbeschränkte k -Band-TM auch $T(n)$ -platzbeschränkt ist, da die Köpfe der k Arbeitsbänder in Zeit $T(n)$ nicht mehr Felder der Arbeitsbänder besuchen können.

Sei M eine k -Band-TM mit Bandalphabet Σ . Dann lässt sich jede **Konfiguration** von M eindeutig beschreiben durch

- 1 Angabe der Position des Lesekopfs auf dem Eingabeband (i.W. n Möglichkeiten),
- 2 Angabe der Bandkonfiguration $\alpha^{(i)}q\beta^{(i)}$, $i = 1, \dots, k$, der k Arbeitsbänder.

Die Länge einer solchen Beschreibung ist, falls M $S(n)$ -platzbeschränkt ist,

$$\leq \lceil \log n + 1 \rceil + k(\lceil \log(S(n) + 1) \rceil + S(n)) + \mathcal{O}(1).$$

Satz 186

Sei die TM M $S(n)$ -platzbeschränkt, $S(n) \geq \log n$. Dann ist die/jede kürzeste terminierende Berechnung von M $c^{S(n)}$ -zeitbeschränkt, für ein geeignetes $c > 0$.

Beweis:

Die Anzahl der **verschiedenen** Konfigurationen einer $S(n)$ -platzbeschränkten k -Band-TM, $S(n) \geq \log n$, ist $\leq c^{S(n)}$ für ein geeignetes $c > 0$, wie sich aus der vorhergehenden Bemerkung ergibt. Da sich bei einer terminierenden DTM bzw. bei einer NDTM in einem kürzesten akzeptierenden Berechnungspfad keine Konfiguration wiederholen kann, folgt daraus die Behauptung.

4. Simulation platzbeschränkter NDTMs

Satz 187 (Satz von Savitch, 1970)

Sei $L \in NSPACE(S(n))$, $S(n) \geq \log n$, $S(n)$ in Platz $(S(n))^2$ berechenbar. Dann ist
$$L \in DSPACE((S(n))^2).$$

Beweis:

Es genügt $L \in DSPACE(\mathcal{O}((S(n))^2))$ zu zeigen.

Unter den gegebenen Voraussetzungen lässt sich jede Konfiguration einer $S(n)$ -platzbeschränkten NDTM N für L in Platz $c \cdot S(n)$, für eine geeignete Konstante (abhängig von N) $c > 0$, darstellen.

Die Länge jeder kürzesten akzeptierenden Berechnung von N , die nur Platz $S(n)$ benötigt, ist beschränkt durch $2^{dS(n)}$, ebenfalls für eine geeignete Konstante $d > 0$.

Beweis (Forts.):

```
proc reach( $C_1, C_2, i$ )  
  if  $i = 0$  and  
    ( $C_1 = C_2$  or  
       $C_2$  in einem Schritt von  $N$  von  $C_1$  aus erreichbar)  
  then return true fi  
  if  $i \geq 1$  then  
    for all Konfigurationen  $C$  der Länge  $c \cdot S(n)$  do  
      if reach( $C_1, C, i - 1$ ) and reach( $C, C_2, i - 1$ ) then  
        return true  
      fi  
    fi  
  return false
```

Beweis (Forts.):

Die Prozedur **reach**, mit Parametern C, C', i , überprüft, ob N , ausgehend von der Konfiguration C , innerhalb von höchstens 2^i Schritten die Konfiguration C' erreichen kann.

Angewandt auf die Anfangskonfiguration und jede mögliche akzeptierende Endkonfiguration von N , mit drittem Parameter $dS(n)$, gestattet diese Prozedur offensichtlich, zu testen, ob N von der Anfangskonfiguration aus und mit der vorgegebenen Platzschränke eine akzeptierende Endkonfiguration erreichen kann.



5. Komplementabschluss von nichtdeterministischem Platz

Wir verwenden in diesem Abschnitt das **strikte** Komplexitätsmaß für die Platzschranke $S(n)$.

Durch die Methode des **induktiven Zählens** zeigen wir

Satz 188

Sei $S(n) \geq \log n$. Dann ist

$$\text{NSPACE}(S(n)) = \text{co-NSPACE}(S(n)).$$

Bemerkung: Für eine Komplexitätsklasse \mathcal{C} ist dabei

$$\text{co-}\mathcal{C} = \{\bar{L}; L \in \mathcal{C}\}.$$

Beweis:

Es genügt zu zeigen

$$L \in \text{NSPACE}(S(n)) \Rightarrow \bar{L} \in \text{NSPACE}(S(n)).$$

Sei $L \in \text{NSPACE}(S(n))$, N eine NDTM für L , w eine Eingabe für N , $|w| = n$. Weiter sei C_w die Menge aller ($S(n)$ -platzbeschränkten) Konfigurationen, die N bei Eingabe w erreichen kann. Es ist

$$|C_w| \leq 2^{dS(n)}$$

für eine geeignete Konstante $d > 0$.

Wir können auch annehmen, dass **jede** Berechnung von N auf w Länge τ hat, für ein $\tau < |C_w|$, und ebenso, dass N eine eindeutige akzeptierende Endkonfiguration $c^{(a)}$ besitzt. Wir modifizieren N so, dass es in der Konfiguration $c^{(a)}$ verbleibt (Endlosschleife), sobald diese Konfiguration erreicht ist.

Beweis (Forts.):

Setze

$C_w(t) :=$ Menge der Konfigurationen, die N nach genau
 t Schritten erreicht, $0 \leq t \leq \tau$

$c_w(t) := |C_w(t)|$

Dann gilt

N akzeptiert w nicht $\Leftrightarrow c^{(a)} \notin C_w(\tau)$

$\Leftrightarrow C_w(\tau)$ enthält $c_w(\tau)$ von $c^{(a)}$

verschiedene Konfigurationen

Wir können daher eine NDTM N' für \bar{L} konstruieren, indem wir nichtdeterministisch einen Beweis für die letzte Aussage raten und verifizieren.

Beweis (Forts.):

Die folgende Prozedur **reach** versucht, bei Eingabe t, cn, c_1, \dots, c_k , nichtdeterministisch zu beweisen, ob $c_i \in C_w(t)$ für mindestens ein $i \in \{1, \dots, k\}$. Ein Berechnungspfad, dem dies nicht gelingt, gibt „?“ zurück. cn steht dabei für $c_w(t)$.

func **reach**(t, cn, c_1, \dots, c_k)

$nc := 0$

for $c \in C_w$ **do**

nichtdeterministisch

tue nichts

oder

rate eine Berechnung von N der Länge t , mit Endkonfiguration c

if erfolgreich **then** $nc := nc + 1$ **fi**

if $c \in \{c_1, \dots, c_k\}$ **then return true fi**

od

if $nc = cn$ **then return false fi**

if $nc < cn$ **then return „?“ fi**

Beweis (Forts.):

Die Prozedur `reach` ist offensichtlich korrekt.

Der Platzbedarf von `reach` ist $\mathcal{O}(S(n))$.

Wir geben nun eine Funktion `count` an, die $c_w(t)$ induktiv berechnet (daher der Name der gesamten Beweistechnik).

Beweis (Forts.):

func count(t, cn)

$nc := 0$

for $c \in C_w$ **do**

bestimme die unmittelbaren Vorgängerkonfigurationen c_1, \dots, c_k von c

$res := \text{reach}(t - 1, cn, c_1, \dots, c_k)$

if $res = \text{true}$ **then** $nc := nc + 1$ **fi**

if $res = \text{„?“}$ **then return** „?“ **fi**

od

return nc

Wird **count** mit $cn = c_w(t - 1)$ aufgerufen und liefert es ein Ergebnis $\neq \text{„?“}$, so ist das Ergebnis gleich $c_w(t)$.

Der Platzbedarf von **count** ist $\mathcal{O}(S(n))$.

Beweis (Forts.):

Damit ergibt sich insgesamt folgende NDTM, die (in Platz $\mathcal{O}(S(n))$) genau alle $w \in \bar{L}$ akzeptiert:

Induktives Zählen:

$$c_w(0) = 1$$

for $t := 1$ **to** τ **do**

if $c_w(t - 1) \neq \text{„?“}$ **then**

$$c_w(t) := \text{count}(t, c_w(t - 1))$$

else

$$c_w(t) := \text{„?“}$$

fi

od

if $c_w(\tau) \neq \text{„?“}$ **then** $res := \text{reach}(\tau, c_w(\tau), c^{(a)})$ **fi**

if $res = \text{false}$ **then** akzeptiere w **fi**

Damit ist $\bar{L} \in \text{NSPACE}(S(n))$. □