Komplexitätstheorie

Wie viele **Ressourcen** (Zeit, Speicherplatz) werden benötigt,

um Sprachen zu entscheiden, Probleme zu lösen ?

Def: $f,g : \mathbb{N} \to \mathbb{R}$

$f \geq_{\text{fü}} g$: für alle n, bis auf endlich viele Ausnahmen gilt $f(n) \geq g(n)$

Def: $g \in \text{FÜP} = \{ h : \mathbb{N} \to \mathbb{R} \mid h \geq_{\text{fü}} 0 \}$

$O(g) = \{ f \in \text{FÜP} \mid \exists c>0 : f \leq_{\text{fü}} c \cdot g \}$

$o(g) = \{ f \in \text{FÜP} \mid \forall c>0 : f \leq_{\text{fü}} c \cdot g \}$

$\Omega(g) = \{ f \in \text{FÜP} \mid \exists c>0 : f \geq_{\text{fü}} c \cdot g \}$

$\omega(g) = \{ f \in \text{FÜP} \mid \forall c>0 : f \geq_{\text{fü}} c \cdot g \}$

$\Theta(g) = O(g) \cap \Omega(g)$

k-Band Turingmaschine:

Eingabeband mit Lesekopf (kein Schreiben)

k Arbeitsbänder, jedes mit einem unabhängigen Lese-/Schreibkopf

$M = (\Sigma, \Gamma, \#, Q, s, F, \Delta)$

Σ	Eingabealphabet
Γ	Arbeitsbandandalphabet
$\#$	Leerzeichen (Inhalt von nicht initialisierten Zellen)
Q	Zustandsmenge (endlich)
$s \in Q$	Startzustand
$F \subset Q$	Endzustände

$\Delta \subset (Q \times \Sigma \times \Gamma^k) \times (Q \times \{L,B,R\} \times (\Gamma \times \{L,B,R\})^k)$ Rechenschrittregeln

Zustand E-Band A-Bänder
derzeitig

Zustand neu, E-Kopf bewegung, zu schreibendes Symbol, A-Kopf-bewegung

1) Eine deterministische Turingmaschine M hat worst-case Laufzeit $f(n)$, wenn M für jedes Eingabewort w nach höchstens $f(|w|)$ Schritten hält.

2) Eine deterministische Turingmaschine M hat worst-case Platzverbrauch $f(n)$, wenn M für jedes Eingabewort w hält und dabei auf jedem Arbeitsband höchstens $f(|w|)$ Speicherzellen verwendet.

3) Eine nicht-deterministische Turingmaschine M hat worst-case Laufzeit $f(n)$, wenn M für jedes Eingabewort $w \in L(M)$ eine akzeptierende Berechnung hat, die nach höchstens $f(|w|)$ Schritten hält.

4) Eine nicht-deterministische Turingmaschine M hat worst-case Platzverbrauch $f(n)$, wenn M für jedes Eingabewort $w \in L(M)$ eine akzeptierende Berechnung hat, die auf jedem Arbeitsband höchstens $f(|w|)$ Speicherzellen verwendet.

$DTIME(f)$ ist die Familie aller Sprachen, die durch irgendeine deterministische Turingmaschine mit worst-case Laufzeit in $O(f)$ entschieden werden können.

$NTIME(f)$ ist die Familie aller Sprachen, die durch irgendeine nicht-deterministische Turingmaschine mit worst-case Laufzeit in $O(f)$ akzeptiert werden können.

$DSPACE(f)$ ist die Familie aller Sprachen, die durch irgendeine deterministische Turingmaschine mit worst-case Platzverbrauch in $O(f)$ entschieden werden können.

$NSPACE(f)$ ist die Familie aller Sprachen, die durch irgendeine nicht-deterministische Turingmaschine mit worst-case Platzverbrauch in $O(f)$ akzeptiert werden können.

Grundlegende Zusammenhänge zwischen Komplexitätsklassen

Grundlemma:

$$\begin{array}{ccc} DTIME(f) & \subseteq & DSPACE(f) \\ \rotatebox{-90}{$\subseteq$} & & \rotatebox{-90}{$\subseteq$} \\ NTIME(f) & \subseteq & NSPACE(f) \end{array}$$

Beweis: jede deterministische TM ist auch eine nicht-deterministische TM

eine TM kann nicht mehr Zellen pro Band verwenden als sie Rechenschritte macht

Hinweis:

TM-Modell: nicht-beschreibbares Eingabeband
k Arbeitsbänder (k unabhängig von Eingabe)

Es wird nur der Platz auf den Arbeitsbändern verrechnet

$\Rightarrow$ sublinearer Platzverbrauch ist möglich

Beispiel: $\{ a^n b^n \mid n \in \mathbb{N} \}$ liegt in $DSPACE(\log n)$

(verwende binären Zähler auf dem Arbeitsband !)

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty} \gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$

Satz A: (Verhältnis zwischen det. Zeit und Platz)

$$\mathrm{DTIME}(f(n)) \subseteq \mathrm{DSPACE}(f(n)) \subseteq \mathrm{DTIME}(2^{\gamma(n) f(n)})$$

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty}\gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$

Satz A: (Verhältnis zwischen det. Zeit und Platz)

$$\mathrm{DTIME}(\, f(n)\,) \subseteq \mathrm{DSPACE}(\, f(n)\,) \subseteq \mathrm{DTIME}(\, 2^{\gamma(n) f(n)}\,)$$

Satz B: (Verhältnis zwischen deterministischer und nicht-det. Zeit)

$$\mathrm{DTIME}(\, f(n)\,) \subseteq \mathrm{NTIME}(\, f(n)\,) \subseteq \mathrm{DTIME}(\, 2^{\gamma(n) f(n)}\,)$$

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty}\gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$ (und so, dass es mit O(f(n)) Platz berechnet werden kann)

Satz A: (Verhältnis zwischen det. Zeit und Platz)

$$\mathrm{DTIME}(\, f(n)\,) \subseteq \mathrm{DSPACE}(\, f(n)\,) \subseteq \mathrm{DTIME}(\, 2^{\gamma(n) f(n)}\,)$$

Satz B: (Verhältnis zwischen deterministischer und nicht-det. Zeit)

$$\mathrm{DTIME}(\, f(n)\,) \subseteq \mathrm{NTIME}(\, f(n)\,) \subseteq \mathrm{DTIME}(\, 2^{\gamma(n) f(n)}\,)$$

Satz C: (Verhältnis zwischen deterministischem und nicht-det. Platz)

$$\mathrm{DSPACE}(\, f(n)\,) \subseteq \mathrm{NSPACE}(\, f(n)\,) \subseteq \mathrm{DSPACE}(\, f^2(n)\,)$$

k-Band Turingmaschine:

Eingabeband mit Lesekopf (kein Schreiben)

k Arbeitsbänder, jedes mit einem unabhängigen Lese-/Schreibkopf

$M = (\Sigma,\Gamma,\#,Q,s,F,\Delta)$

Σ	Eingabealphabet
Γ	Arbeitsbandandalphabet
$\#$	Leerzeichen (Inhalt von nicht initialisierten Zellen)
Q	Zustandsmenge (endlich)
$s \in Q$	Startzustand
$F \subset Q$	Endzustände

$\Delta \subset (Q\times\Sigma\times\Gamma^k) \times (Q\times\{L,B,R\}\times(\Gamma\times\{L,B,R\})^k)$ Rechenschrittregeln

Zustand E-Band A-Bänder
d e r z e i t i g

Zustand neu, E-Kopf bewegung, zu schreibendes Symbol, A-Kopf-bewegung

Konfiguration von TM M

Inhalt Eingabeband	Lesekopf Position	Zustand	Inhalt der k Arbeitsbänder	Positionen der k Lese/Schreibköpfe
Σ^* $\times$	$\mathbb{N}$ $\times$	Q $\times$	$(\Gamma^*)^k$ $\times$	$\mathbb{N}^k$

Konfiguration von TM M

Inhalt Eingabeband		Lesekopf Position		Zustand		Inhalt der k Arbeitsbänder		Positionen der k Lese/Schreibköpfe
Σ^*	$\times$	$\mathbb{N}$	$\times$	Q	$\times$	$(\Gamma^*)^k$	$\times$	$\mathbb{N}^k$

Konfigurationsgraph von TM M

Knoten: Konfigurationen $\Sigma^* \times \mathbb{N} \times Q \times (\Gamma^*)^k \times \mathbb{N}^k$

Kanten: $[C,C']$ wenn **M** in einem Rechenschritt von Konf. C zu Konfig. C' kommt

Konfiguration von TM M

Inhalt Eingabeband		Lesekopf Position		Zustand		Inhalt der k Arbeitsbänder		Positionen der k Lese/Schreibköpfe
Σ^*	$\times$	$\mathbb{N}$	$\times$	Q	$\times$	$(\Gamma^*)^k$	$\times$	$\mathbb{N}^k$

Konfigurationsgraph von TM M

Knoten: Konfigurationen $\Sigma^* \times \mathbb{N} \times Q \times (\Gamma^*)^k \times \mathbb{N}^k$

Kanten: $[C,C']$ wenn **M** in einem Rechenschritt von Konf. C zu Konfig. C' kommt

TM M akzeptiert Eingabe w $\Leftrightarrow$ $\exists$ gerichteten Pfad von $start_w$ zu einer Endkonfiguration

Konfiguration von TM M bei Eingabe w und höchstens N Zellen/A-Band

Inhalt Eingabeband		Lesekopf Position		Zustand		Inhalt der k Arbeitsbänder		Positionen der k Lese/Schreibköpfe
$\{w\}$	$\times$	$\{1,\cdots,\lvert w\rvert\}$	$\times$	Q	$\times$	$(\Gamma^N)^k$	$\times$	$\{1,\cdots,N\}^k$

$KG_M(w,N)$ **Konfigurationsgraph von TM M** bei Eingabe w und höchstens N Zellen/A-Band

Knoten: Konfigurationen $\{1,\cdots,\lvert w\rvert\} \times Q \times (\Gamma^N)^k \times \{1,\cdots,N\}^k$

Kanten: $[C,C']$ wenn **M** in einem Rechenschritt von Konf. C zu Konfig. C' kommt

Anzahl der Knoten: $\leq$ $n \cdot \lvert Q\rvert \cdot \lvert\Gamma\rvert^{kN} \cdot N^k$ — $n=\lvert w\rvert$

Anzahl der Kanten: $\leq$ $\alpha \cdot n \cdot \lvert Q\rvert \cdot \lvert\Gamma\rvert^{kN} \cdot N^k$ — α eine von M abh. Konstante

$KG_M(w,N)$ **Konfigurationsgraph von TM M** bei Eingabe w und höchstens N Zellen/A-Band

$n=\lvert w\rvert$

Knoten: Konfigurationen

Kanten: $[C,C']$ wenn **M** in einem Rechenschritt von Konf. C zu Konfig. C' kommt

Anzahl der Knoten: $\leq$ $n \cdot \lvert Q\rvert \cdot \lvert\Gamma\rvert^{kN} \cdot N^k$ $\leq$ A^N (falls $N \geq \log_2 n$) — A eine von M abh. Konstante

Eingrad und Ausgrad jedes Knoten $\leq$ α — α eine von M abh. Konstante

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty}\gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$

Satz A: (Verhältnis zwischen det. Zeit und Platz)

$$\mathrm{DTIME}(\ f(n)\ ) \subseteq \mathrm{DSPACE}(\ f(n)\ ) \subseteq \mathrm{DTIME}(\ 2^{\gamma(n)f(n)}\ )$$

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty}\gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$

Satz A: (Verhältnis zwischen det. Zeit und Platz)

$$\mathrm{DTIME}(\ f(n)\ ) \subseteq \mathrm{DSPACE}(\ f(n)\ ) \subseteq \mathrm{DTIME}(\ 2^{\gamma(n)f(n)}\ )$$

Lemma A: (i) $L \in \mathrm{DTIME}(\ f(n)\ ) \Rightarrow L \in \mathrm{DSPACE}(\ f(n)\ )$

(ii) $L \in \mathrm{DSPACE}(\ f(n)\ ) \Rightarrow L \in \mathrm{DTIME}(\ A^{f(n)}\ )$

für eine von L abh.
Konstante A

Sei $f(n) \geq \log_2 n$

Lemma A: (i) $L \in \mathrm{DTIME}(\ f(n)\ ) \Rightarrow L \in \mathrm{DSPACE}(\ f(n)\ )$

(ii) $L \in \mathrm{DSPACE}(\ f(n)\ ) \Rightarrow L \in \mathrm{DTIME}(\ A^{f(n)}\ )$

für eine von L abh.
Konstante A

Bew: (i) trivial
(ii) M det. TM für L mit Platzverbrauch $f(n)$
w Eingabe für M $n=|w|$, $N=f(n)$ ($\geq \log_2 n$)

Berechnung von M entspricht Pfad durch $KG_M(w,N)$

$KG_M(w,N)$ hat höchstens A^N Knoten

$\Rightarrow$ M macht auf Eingabe w höchstens A^N viele Schritte (sonst Endlosloop !)

$\Rightarrow$ M entscheidet L in Zeit $A^{f(n)}$ $\Rightarrow$ $L \in \mathrm{DTIME}(\ A^{f(n)}\ )$

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty}\gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$ (und so, dass es mit $O(f(n))$ Platz berechnet werden kann)

Satz B: (Verhältnis zwischen deterministischer und nicht-det. Zeit)

$$\mathrm{DTIME}(\ f(n)\ ) \subseteq \mathrm{NTIME}(\ f(n)\ ) \subseteq \mathrm{DTIME}(\ 2^{\gamma(n)f(n)}\ )$$

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty}\gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$ (und so, dass es mit O(f(n)) Platz berechnet werden kann)

Satz B: (Verhältnis zwischen deterministischer und nicht-det. Zeit)

$$\mathrm{DTIME}(\, f(n)\,) \subseteq \mathrm{NTIME}(\, f(n)\,) \subseteq \mathrm{DTIME}(\, 2^{\gamma(n)f(n)}\,)$$

Lemma B: (i) $L \in \mathrm{DTIME}(\, f(n)\,) \Rightarrow L \in \mathrm{NTIME}(\, f(n)\,)$

(ii) $L \in \mathrm{NTIME}(\, f(n)\,) \Rightarrow L \in \mathrm{DTIME}(\, A^{f(n)}\,)$

für eine von L abh.
Konstante A

Sei $f(n) \geq \log_2 n$

Lemma B: (i) $L \in \mathrm{DTIME}(\, f(n)\,) \Rightarrow L \in \mathrm{NTIME}(\, f(n)\,)$

(ii) $L \in \mathrm{NTIME}(\, f(n)\,) \Rightarrow L \in \mathrm{DTIME}(\, A^{f(n)}\,)$

für eine von L abh.
Konstante A

Bew: (i) trivial

(ii) M nicht-det. TM für L mit Zeitverbrauch und daher Platzverbrauch $f(n)$
w Eingabe für M $\quad n=|w|,\ N=f(n) \quad (\geq \log_2 n)$

Berechnung von M entspricht Pfad durch $KG_M(w,N)$
von $init(w)$ zu einer Endkonfiguration

$KG_M(w,N)$ hat höchstens A^N Knoten und $O(A^N)$ Kanten

$\Rightarrow$ det. TM M' macht Tiefensuche (DFS) in $KG_M(w,N)$ und testet, ob
eine Endkonfiguration von $start_w$ erreichbar; braucht Zeit $O(\#\text{ Kanten})$

$\Rightarrow$ M' entscheidet L in Zeit $O(A^{f(n)}) \Rightarrow L \in \mathrm{DTIME}(\, A^{f(n)}\,)$

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty}\gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$ (und so, dass es mit O(f(n)) Platz berechnet werden kann)

Satz C: (Verhältnis zwischen deterministischem und nicht-det. Platz)

$$\mathrm{DSPACE}(\, f(n)\,) \subseteq \mathrm{NSPACE}(\, f(n)\,) \subseteq \mathrm{DSPACE}(\, f^2(n)\,)$$

Sei $\gamma : \mathbb{N} \to \mathbb{R}$ beliebig langsam wachsend,
aber nicht-fallend und $\lim_{n\to\infty}\gamma(n) = \infty$
z.B. $\gamma(n) = \log\log\log\log\log n$

Sei $f(n) \geq \log_2 n$ (und so, dass es mit O(f(n)) Platz berechnet werden kann)

Satz C: (Verhältnis zwischen deterministischem und nicht-det. Platz)

$$\mathrm{DSPACE}(\, f(n)\,) \subseteq \mathrm{NSPACE}(\, f(n)\,) \subseteq \mathrm{DSPACE}(\, f^2(n)\,)$$

Lemma C: (i) $L \in \mathrm{DSPACE}(\, f(n)\,) \Rightarrow L \in \mathrm{NSPACE}(\, f(n)\,)$

(ii) $L \in \mathrm{NSPACE}(\, f(n)\,) \Rightarrow L \in \mathrm{DSPACE}(\, f(n)^2\,)$

Lemma C: (i) $L \in \mathrm{DSPACE}(\ f(n)\ ) \Rightarrow L \in \mathrm{NSPACE}(\ f(n)\ )$

(ii) $L \in \mathrm{NSPACE}(\ f(n)\ ) \Rightarrow L \in \mathrm{DSPACE}(\ f(n)^2\ )$

Bew: (i) trivial

(ii) M nicht-det. TM für L mit daher Platzverbrauch f(n)

w Eingabe für M $n=|w|$, $N=f(n)$ ($\geq \log_2 n$)

Berechnung von M entspricht Pfad durch $KG_M(w,N)$ von init(w) zu einer Endkonfiguration

$KG_M(w,N)$ hat höchstens A^N Knoten und $O(A^N)$ Kanten

$\Rightarrow$ Brauche det. TM M' (deterministischen Algorithmus) zum Testen, ob in $KG_M(w,N)$ eine Endkonfiguration von $start_w$ erreichbar ist;

Dieser Algorithmus soll wenig Platz brauchen !!

Der Beweis reduziert sich auf "Berechnen" von Erreichbarkeit in einem sehr großen, implizit gegebenen Graphen. Es muss die Frage beantwortet werden

$\exists$ gerichteter Pfad von $start_w$ zu einer Endkonfiguration im Konfigurationsgraphen $KG_M(w,N)$

$n=|w|$ $N=f(n)$

Abstraktes Problem:

Gerichteter Graph $G = (V,E)$ ist implizit gegeben, d.h. man kann
(i) die Knoten in V aufzählen
(ii) für zwei Knoten u,v testen, ob u=v
(iii) für zwei Knoten u,v testen, ob $\langle u,v\rangle \in E$
und zwar jeweils mit Speicherverbrauch $O(\log |V|)$ (Knotendarstellungsgröße)

Berechne

erreichbar(s , v, λ) . . . gibt es in G einen gerichteten Pfad von s nach v der Länge höchstens λ ?

Für Beweis von Lemma C, deterministische Lösung mit $O(\log^2|V|)$ Platzverbrauch

```
erreichbar( s , v, λ )  =  if λ =0  then return (s=v)
                           if λ =1  then return (s=v) or [s,v⟩∈E
                           else
                              for each m∈V do
                                 if    erreichbar( s , m , ⌊λ/2⌋ )
                                   and erreichbar( m , v ,⌈λ /2⌉ )
                                 then return true
                              endfor
                              return false
```

Laufzeit: $O((\log \lambda)\cdot \text{Darstellungsgröße}(v)) = O(\log^2|V|)$ mit $\lambda=|V|$

Für Beweis von Lemma C, deterministische Lösung mit $O(\log^2|V|)$ Platzverbrauch

```
erreichbar( s , v, λ )  =  if λ =0  then return (s=v)
                           if λ =1 then return (s=v) or [s,v⟩∈E
                           else
                               for each m∈V do
                                   if      erreichbar( s , m , ⌊λ/2⌋ )
                                      and     erreichbar( m , v ,⌈λ /2⌉ )
                                   then return true
                               endfor
                               return false
```

Laufzeit: $O((\log \lambda)\cdot$ Darstellungsgröße$(v)) = O(\log^2|V|)$ mit $\lambda=|V$

da die beiden rekursiven Aufrufe hintereinander den gleichenPlatz verwenden können|