Rekursionssatz $t : \Sigma^* \times \Sigma^* \to \Sigma^*$ von TM T berechnet

(Eingabe hat Form $u\$v$)

$\exists$ TM R, die die Funktion $r : \Sigma^* \to \Sigma^*$ mit

$r(x) = t(\langle R \rangle, x)$

berechnet.

Betrachte Sprache MIN-TM $= \{ u \in \{0,1\}^* \mid \forall v \in \{0,1\}^* : L(M_u)=L(M_v) \Rightarrow |v| \geq |u| \}$

$u \in$ MIN-TM bedeutet u ist eine ***"kürzeste"*** TM-Kodierung für die Sprache $L=L(M_u)$

Satz: MIN-TM ist **nicht** entscheidbar.

Beweis: Nimm das Gegenteil an, und es sei **E** eine TM, die MIN-TM entscheidet.

Betrachte TM **A**, die mit Eingabe x Folgendes macht:

1. Bestimme $w = \langle A \rangle$ (möglich wegen des Rekursionssatzes)
2. for $k = |w|+1$ to ∞ do
   for each $u \in \{0,1\}^k$ do teste mit E, ob $u \in$ MIN-TM
   wenn ja, gehe zu 3.
3. Simuliere M_u auf x

$|$MIN-TM$| = \infty$, aber Schritt 2 lässt nur endlich viele u's aus. D.h.Schritt 3 wird erreicht.

$L(A) = L(M_u)$, aber $|\langle A \rangle| < |u|$, also Widerspruch dass $u \in$ MIN-TM.

Fixpunktsatz:
Sei $t : \Sigma^* \to \Sigma^*$ eine überall berechenbare Funktion

Dann gibt es ein $u \in \{0,1\}^*$ mit $L(M_u) = L(M_{t(u)})$.

Fixpunktsatz:
Sei $t : \Sigma^* \to \Sigma^*$ eine überall berechenbare Funktion

Dann gibt es ein $u \in \{0,1\}^*$ mit $L(M_u) = L(M_{t(u)})$.

Beweis: Sei F eine TM mir folgendem Verhalten:

Eingabe x
1. Sei u die Kodierung von F (möglich nach Rekursionssatz)
2. Berechne $w = t(u)$
3. Simuliere M_w auf Eingabe x

Dann gilt: F verhält sich genau wie M_w, also $L(F) = L(M_w)$

Aber $F = M_u$ und $w = t(u)$

also $L(M_u) = L(M_{t(u)})$

Komplexitätstheorie

Wie viele **Ressourcen** (Zeit, Speicherplatz) werden benötigt, um Sprachen zu entscheiden, Probleme zu lösen ?

Def: $f,g : \mathbb{N} \to \mathbb{R}$
$f \geq_{fü} g$: für alle n, bis auf endlich viele Ausnahmen gilt $f(n) \geq g(n)$

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

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

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

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

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

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

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 Zustand E-Kopf zu schreibendes A-Kopf-
d e r z e i t i g neu bewegung Symbol 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: $\text{DTIME}(f) \subseteq \text{DSPACE}(f)$

$\text{DTIME}(f) \subseteq \text{NTIME}(f)$, $\text{DSPACE}(f) \subseteq \text{NSPACE}(f)$

$\text{NTIME}(f) \subseteq \text{NSPACE}(f)$

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)

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

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)

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

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

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

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)

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

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

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

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

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

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	$\times$	$(\Gamma^*)^k$	$\times$	$\mathbb{N}^k$

Konfigurationsgraph von TM M

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

Kanten: $[C,C'\rangle$ 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
$\{w\}$	$\times$	$\{1,\cdots,\|w\|\}$	$\times$	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 $\{1,\cdots,|w|\} \times Q \times (\Gamma^N)^k \times \{1,\cdots,N\}^k$

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

Anzahl der Knoten: $\leq$ $n \cdot |Q| \cdot |\Gamma|^{kN} \cdot N^k$ $n=|w|$
Anzahl der Kanten: $\leq$ $\alpha \cdot n \cdot |Q| \cdot |\Gamma|^{kN} \cdot N^k$ α eine von M abh. Konstante

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

Knoten: Konfigurationen
Kanten: $[C,C'\rangle$ wenn **M** in einem Rechenschritt von Konf. C zu Konfig. C' kommt

Anzahl der Knoten: $\leq$ $n \cdot |Q| \cdot |\Gamma|^{kN} \cdot N^k$ $\leq$ 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