Algorithmische Komplexitätstheorie

Prof. Dr. Egon Wanke

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
Algorithmische Komplexitätstheorie

Mehrband-Turingmaschinen mit Ein- und Ausgabe (Nr. 61)

7. Mai 2021

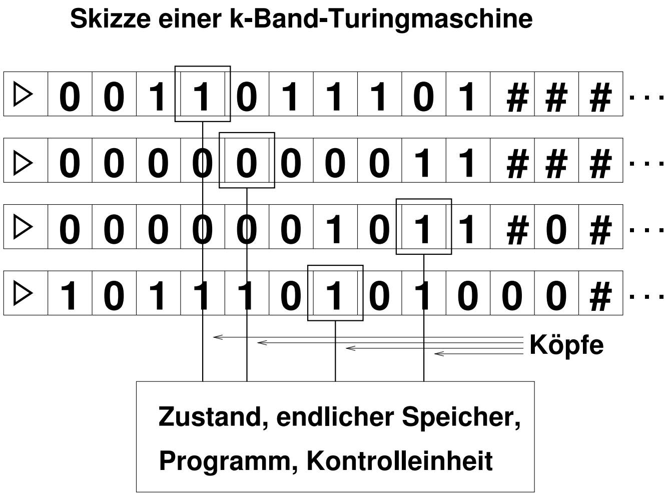
k-Band-Turingmaschine

Definition

Eine *k-Band-Turingmaschine* ist eine System $M = (Q, \Sigma, \delta, s)$, wobei Q, Σ und s wie für eine 1-Band-Turingmaschine definiert sind und

$$\delta : Q \times \Sigma^k \longrightarrow (Q \cup \{h, h_y, h_n\}) \times (\Sigma \times \{\rightarrow, \leftarrow, \perp\})^k$$

das Programm von M ist.


Skizze einer k-Band-Turingmaschine
▷ 0 0 1 1 0 1 1 1 0 1 # # # ...
▷ 0 0 0 0 0 0 0 0 1 1 # # # ...
▷ 0 0 0 0 0 0 1 0 1 1 # 0 # ...
▷ 1 0 1 1 1 0 1 0 1 0 0 0 # ...
Köpfe
Zustand, endlicher Speicher,
Programm, Kontrolleinheit

Unterschiede zur 1-Band und k-Band-Turingmaschinen

- Eine k-Band-Turingmaschine hat k Bänder und k Köpfe.
- Sei $\delta(q, a_1, \ldots, a_k) = (q', b_1, D_1, \ldots, b_k, D_k)$.
  Wenn M im Zustand q ist und Kopf i, $1 \leq i \leq k$, auf Band i das Symbol a_i liest, wechselt die Maschine in den Zustand q' und überschreibt a_i mit b_i und bewegt Kopf i in Richtung D_i.
- Die Eingabe steht auf dem ersten Band, die Ausgabe im Zustand h steht auf dem letzten Band.
- Die Köpfe bewegen sich niemals nach links über die initialen Symbole hinaus. Die initialen Symbole werden niemals überschrieben.
- Konfigurationen sind $(1 + 2 \cdot k)$-Tupel.
- Eine k-Band-Turingmaschine startet in der Konfiguration (*Startkonfiguration*)

$$\beta_0 = (s, \triangleright, x, \triangleright, \varepsilon, \cdots \triangleright, \varepsilon).$$

Unterschiede zur 1-Band und k-Band-Turingmaschinen

- Die Relationen $\overset{M}{\longmapsto}$, $\overset{M^k}{\longmapsto}$, $\overset{M^*}{\longmapsto}$ sind analog zur 1-Band-Turingmaschinen definiert. (Übungsaufgabe!)
- $M(x) = \text{yes}$, falls $\beta_0 \overset{M^*}{\longmapsto} (h_y, u_1, w_1, \ldots, u_k, w_k)$ für $u_1, w_1, \ldots, u_k, w_k \in \Sigma^*$.
- $M(x) = \text{no}$, falls $\beta_0 \overset{M^*}{\longmapsto} (h_n, u_1, w_1, \ldots, u_k, w_k)$ für $u_1, w_1, \ldots, u_k, w_k \in \Sigma^*$.
- $M(x) = y \in \Sigma^*$, falls $\beta_0 \overset{M^*}{\longmapsto} (h, u_1, w_1, \ldots, u_k, w_k)$ für $u_1, w_1, \ldots, u_k, w_k \in \Sigma^*$ und y das Wort $u_k w_k$ ist ohne führendes initiales Symbol und ohne die rechten Leerzeichen.
- $M(x) = \nearrow$, falls M auf Eingabe x nicht hält.

Beispiel

Sei $M = (Q, \Sigma, \delta, s)$, $Q = \{s, p, q\}$, und $\Sigma = \{0, 1, \triangleright, \#\}$

q	a_1	a_2	$\delta(q, a_1, a_2)$		
s	0	$\#$	s	$(0, \rightarrow)$	$(0, \rightarrow)$
s	1	$\#$	s	$(1, \rightarrow)$	$(1, \rightarrow)$
s	$\triangleright$	$\triangleright$	s	$(\triangleright, \rightarrow)$	$(\triangleright, \rightarrow)$
s	$\#$	$\#$	q	$(\#, \leftarrow)$	$(\#, \perp)$
q	0	$\#$	q	$(0, \leftarrow)$	$(\#, \perp)$
q	1	$\#$	q	$(1, \leftarrow)$	$(\#, \perp)$
q	$\triangleright$	$\#$	p	$(\triangleright, \rightarrow)$	$(\#, \leftarrow)$
p	0	0	p	$(0, \rightarrow)$	$(0, \leftarrow)$
p	0	1	h_n	$(0, \perp)$	$(1, \perp)$
p	1	0	h_n	$(1, \perp)$	$(0, \perp)$
p	1	1	p	$(1, \rightarrow)$	$(1, \leftarrow)$
p	$\#$	$\triangleright$	h_y	$(\#, \perp)$	$(\triangleright, \perp)$

k-Band-Turingmaschine

Beispiel

	Band 1							Band 2						
s	$\underline{\triangleright}$	0	1	0	1	0		$\underline{\triangleright}$						
s	$\triangleright$	$\underline{0}$	1	0	1	0		$\triangleright$	$\underline{\#}$					
s	$\triangleright$	0	$\underline{1}$	0	1	0		$\triangleright$	0	$\underline{\#}$				
s	$\triangleright$	0	1	$\underline{0}$	1	0		$\triangleright$	0	1	$\underline{\#}$			
s	$\triangleright$	0	1	0	$\underline{1}$	0		$\triangleright$	0	1	0	$\underline{\#}$		
s	$\triangleright$	0	1	0	1	$\underline{0}$		$\triangleright$	0	1	0	1	$\underline{\#}$	
s	$\triangleright$	0	1	0	1	0	$\underline{\#}$	$\triangleright$	0	1	0	1	0	$\underline{\#}$
q	$\triangleright$	0	1	0	1	$\underline{0}$	#	$\triangleright$	0	1	0	1	0	$\underline{\#}$
q	$\triangleright$	0	1	0	$\underline{1}$	0	#	$\triangleright$	0	1	0	1	0	$\underline{\#}$
q	$\triangleright$	0	1	$\underline{0}$	1	0	#	$\triangleright$	0	1	0	1	0	$\underline{\#}$
q	$\triangleright$	0	$\underline{1}$	0	1	0	#	$\triangleright$	0	1	0	1	0	$\underline{\#}$
q	$\triangleright$	$\underline{0}$	1	0	1	0	#	$\triangleright$	0	1	0	1	0	$\underline{\#}$
q	$\underline{\triangleright}$	0	1	0	1	0	#	$\triangleright$	0	1	0	1	0	$\underline{\#}$
p	$\triangleright$	$\underline{0}$	1	0	1	0	#	$\triangleright$	0	1	0	1	$\underline{0}$	#
p	$\triangleright$	0	$\underline{1}$	0	1	0	#	$\triangleright$	0	1	0	$\underline{1}$	0	#
p	$\triangleright$	0	1	$\underline{0}$	1	0	#	$\triangleright$	0	1	$\underline{0}$	1	0	#
p	$\triangleright$	0	1	0	$\underline{1}$	0	#	$\triangleright$	0	$\underline{1}$	0	1	0	#
p	$\triangleright$	0	1	0	1	$\underline{0}$	#	$\triangleright$	$\underline{0}$	1	0	1	0	#
p	$\triangleright$	0	1	0	1	0	$\underline{\#}$	$\underline{\triangleright}$	0	1	0	1	0	#
h_y	$\triangleright$	0	1	0	1	0	$\underline{\#}$	$\underline{\triangleright}$	0	1	0	1	0	#

k-Band-Turingmaschine

Definition

Sei $M = (Q, \Sigma, \delta, s)$ eine k-Band-Turingmaschine,
$x \in (\Sigma - \{\#, \triangleright\})^*$ eine Eingabe für M und $f : \mathbb{N}_0 \to \mathbb{N}_0$ ein Funktion.

- M berechnet $M(x)$ in t Schritten, falls
  $\beta_0 = (s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon,) \xmapsto{M^t} (q, u_1, w_1, \ldots, u_k, w_k)$ mit $q \in \{h, h_y, h_n\}$.
  (Falls $M(x) = \nearrow$, dann ist die Anzahl der Schritte unendlich.)
- M ist $f(n)$ *zeitbeschränkt*, falls für jede Eingabe $x \in (\Sigma - \{\#, \triangleright\})^*$ mit $|x| = n$, $M(x)$ in höchstens $f(n)$ Schritten berechnet wird.
- $f(n)$ ist eine *Zeitschranke* für M.

Komplexitätsklassen

Definition

$\mathrm{TIME}(f(n))$ ist die Klasse aller Sprachen $L \subseteq (\Sigma - \{\triangleright, \#\})^*$, die mit einer Mehrband-Turingmaschine mit Zeitschranke $f(n)$ entschieden werden können.

$\mathrm{TIME}(f(n))$ ist eine *Komplexitätsklasse*.

Anmerkungen

1. Sei L_p die Sprache alle Palindrome.
2. Unsere 1-Band-Turingmaschine zur Erkennung von Palindromen benötigte $(2n+1)+(2n-3)+(2n-7)+\dots$ Schritte um festzustellen, ob ein Wort der Länge n ein Palindrom ist oder nicht.
3. Somit ist $L_p \in \mathrm{TIME}\left(\frac{(n+1)(n+2)}{2}\right)$.
4. Unsere 2-Band-Turingmaschine zur Erkennung von Palindromen benötigte $f'(n) = 3n+4$ Schritte um festzustellen, ob ein Wort der Länge n ein Palindrom ist oder nicht.
5. Somit ist $L_p \in \mathrm{TIME}(3n+4)$.

Simulation von Turingmaschinen

Satz

Für jede k-Band-Turingmaschine $M = (Q, \Sigma, \delta, s)$ *mit Zeitschranke* $f(n) \geq n$ *gibt es eine 1-Band-Turingmaschine* $M' = (Q', \Sigma', \delta', s')$ *mit Zeitschranke* $g(n) \in O(f(n)^2)$*, so dass* $\forall\ x \in (\Sigma - \{\#, \triangleright\})^* : M(x) = M'(x)$.

Beweis.

Wir simulieren mit einer 1-Band-Turingmaschine M' *alle* k *Bänder von* M *auf einem Band.*

Sei $\Sigma'' = \Sigma \cup \{\triangleleft, \triangleright', \triangleleft', \#'\}$*, sei* $\underline{\Sigma''} = \{\underline{a} \mid a \in \Sigma''\}$*, und sei* $\Sigma' = \Sigma'' \cup \underline{\Sigma''}$.

Simulation von Turingmaschinen

Beweis Fortsetzung.

Eine Konfiguration

$$(q, u_1, w_1, \ldots, u_k, w_k)$$

für M entspricht einer Konfiguration

$$(q', \triangleright, u_1' w_1 \triangleleft' \; u_2' w_2 \triangleleft' \ldots u_k' w_k \triangleleft' \triangleleft)$$

für M', wobei u_i' das Wort u_i ist, in dem $\triangleright$ durch $\triangleright'$ und das letzte Symbol a_i durch $\underline{a_i}$ ersetzt wurde.
Das letzte Symbol $\triangleleft$ kennzeichnet das Bandende für M'.

1. *M' ersetzt die Eingabe x durch $\triangleright' x \triangleleft' (\triangleright' \triangleleft')^{k-1} \triangleleft$.*
   Dies kann mit zusätzlichen $O(k + |\Sigma|)$ Zuständen erreicht werden.

Simulation von Turingmaschinen

Beweis Fortsetzung.

1. *Zur Simulation eines Schrittes der k-Band-Turingmaschine M läuft M' von $\triangleright$ nach $\triangleleft$ und zurück und merkt sich dabei im aktuellen Zustand den Zustand von M und die Symbole der k Köpfe.*
   Dies kann mit zusätzlichen $O(|Q| \cdot |\Sigma|^k)$ Zuständen erreicht werden.
2. *M' kann nun von links nach rechts an den Kopfpositionen die entsprechenden (maximal zwei) Symbole ändern. Muß auf einem Band ein Leerzeichen eingeführt werden, so wird $\triangleleft'$ mit $\#'$ überschrieben, alles rechts von $\#'$ eine Position nach rechts verschoben, $\#'$ mit $\underline{\#}$ überschrieben und das Symbol rechts von $\#'$ mit $\triangleleft'$ überschrieben.*
   Dies kann mit einer Zustandserweiterung um einen Faktor aus $O(k + |\Sigma|)$ erreicht werden.

Beweis Fortsetzung.

1. *Wenn M hält, ersetzt M' die Ausgabe $\rhd' y_1 \lhd' \rhd' y_2 \lhd' \cdots \rhd' y_k \lhd' \lhd$ von M durch y_k und hält im Zustand h.*

   Dies kann mit zusätzlichen $O(|\Sigma|)$ Zuständen erreicht werden.

Da M in Zeit $f(n)$ arbeitet, werden die k Wörter nicht länger als $f(|x|) + 1$ und somit ist das Wort auf dem Band von M' nicht länger als $k \cdot (f(|x|) + 2) + 2$.

M' läuft für jeden Schritt von M zweimal von links nach rechts und zurück und höchstens einmal zusätzlich nach rechts und zurück für jedes Leerzeichen, das eingefügt werden muß.

Daraus resultiert für jeden Simulationsschritt (für konstantes k) eine Laufzeit aus $O(f(|x|))$ und somit eine totale Laufzeit aus $O(f(|x|)^2)$. □

Speedup-Theorem

Satz

Sei $L \in TIME(f(n))$. Für jedes $\epsilon > 0$ ist $L \in TIME(\epsilon \cdot f(n) + O(n))$.

Beweis.

Sei $M = (Q, \Sigma, \delta, s)$ eine k-Band-Turingmaschine, die L in Zeit $f(n)$ entscheidet.

Wir konstruieren eine k'-Band-Turingmaschine $M' = (Q', \Sigma', \delta', s')$, die L in Zeit $f'(n) = \epsilon \cdot f(n) + O(n)$ entscheidet. Für $k = 1$ sei $k' = 2$, ansonsten sei $k' = k$.

<u>Idee:</u> Wir kodieren mehrere Symbole aus Σ mit einem Symbol in Σ'. Dadurch kann M' in einer festen Anzahl von Schritten beliebig viele Schritte von M simulieren.

Sei r die Anzahl der Symbole, die zusammengefaßt werden sollen, und $\Sigma' = \Sigma \cup (\Sigma \cup \underline{\Sigma})^r$.

Speedup-Theorem

Beweis Fortsetzung.

Sei $\Sigma = \{0, 1, \#, \triangleright\}$ *und* $r = 4$.
Eine Konfiguration

$$(q,\ \triangleright\, x_1\ x_2\ x_3,\ \ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ \#\ \ \ldots)$$

mit $x_i \in \Sigma$ *für* M *entspricht einer Konfiguration*

$$(q', \triangleright\, (x_1, x_2, \underline{x}_3, x_4),\ (x_5, x_6, x_7, x_8)\ (x_9, \#, \#, \#))$$

für M'.

Speedup-Theorem

Beweis Fortsetzung.

M	M'
$q_1 \quad \triangleright\ x_1\ x_2\ \underline{x_3}\ x_4\ x_5\ x_6\ x_7\ x_8$	$q_1' \quad \triangleright\ \underline{(x_1, x_2, \underline{x_3}, x_4)}\ (x_5, x_6, x_7, x_8)$
$q_2 \quad \triangleright\ x_1\ \underline{x_2}\ y_3\ x_4\ x_5\ x_6\ x_7\ x_8$	
$q_3 \quad \triangleright\ x_1\ y_2\ \underline{y_3}\ x_4\ x_5\ x_6\ x_7\ x_8$	
$q_4 \quad \triangleright\ x_1\ y_2\ z_3\ \underline{x_4}\ x_5\ x_6\ x_7\ x_8$	
$q_5 \quad \triangleright\ x_1\ y_2\ z_3\ y_4\ \underline{x_5}\ x_6\ x_7\ x_8$	$q_5' \quad \triangleright\ (x_1, y_2, z_3, y_4)\ \underline{(\underline{x_5}\ x_6, x_7, x_8)}$

M' schreibt zuerst die Eingabe x in umgewandelter Form auf das zweite Band.

Eingabe 0110100101 *wird zum Beispiel für* $r = 4$ *transformiert in*

$$(0110)(1001)(01\#\#).$$

Speedup-Theorem

Beweis Fortsetzung.

Das zweite Band wird jetzt als Eingabeband betrachtet.

M' simuliert in höchstens 6 Schritten r Schritte von M. Dazu bewegt M' alle Köpfe eine Position nach rechts, zwei Positionen nach links und wieder eine Position nach rechts.

Dabei merkt sich M' im Zustand alle Symbole links und rechts von den Köpfen und kann nun in 2 weiteren Schritten r Schritte von M simulieren. (r Schritte von M können maximal 2 Symbole pro Band für M' verändern.)

Wenn M hält, dann hält auch M'.

M' hält auf Eingabe x nach $O(|x|) + 6\lceil \frac{f(|x|)}{r} \rceil$ Schritten.

Mit $r \geq \lceil \frac{6}{\epsilon} \rceil$, folgt $L \in TIME(\epsilon \cdot f(n) + O(n))$. $\square$

Platzbeschränkte Berechnungen

Sei

$$(s, \triangleright, x, \triangleright, \epsilon, \ldots, \triangleright, \epsilon) \overset{M^*}{\longmapsto} (H, u_1, w_1, \ldots)$$

mit $H \in \{h, h_y, h_n\}$ eine Berechnung.

Sei $\sum_{i=1}^{k} \mid u_i w_i \mid$ der für die Berechnung insgesamt benötigte Platz.

Dann können Palindrome x auf Platz $O(|x|)$ entschieden werden.

Die Definition einer Platzschranke ohne Berücksichtigung der Eingabe und Ausgabe kann jedoch zu besseren Ergebnissen führen.

Palindrome können dann zum Beispiel auf Platz $O(\log(n))$ erkannt werden.

Definition

Eine *k-Band-Turingmaschine mit Ein- und Ausgabe* ist eine k-Band-Turingmaschine $M = (Q, \Sigma, \delta, s)$, in der das Programm δ wie folgt eingeschränkt ist. Sei $\delta(q, a_1, \ldots, a_k) = (q', b_1, D_1, \ldots, b_k, D_k)$, dann gilt

- $a_1 = b_1$ (das erste Band wird nie verändert),
- Falls $a_1 = \#$, dann ist $D_1 \in \{\leftarrow, \perp\}$ (die Eingabe wird nicht verlassen) und
- $D_k \in \{\perp, \rightarrow\}$ (auf dem letzten Band kann links vom Kopf nicht korrigiert werden).

Platzbeschränkte Berechnungen

Definition

Sei $M = (Q, \Sigma, \delta, s)$ eine k-Band-Turingmaschine mit Ein- und Ausgabe, $x \in (\Sigma - \{\#, \triangleright\})^*$ eine Eingabe für M und $f : \mathbb{N}_0 \to \mathbb{N}_0$ ein Funktion.

- M berechnet $M(x)$ auf Platz t, falls $\beta_0 = (s, \triangleright, x, \triangleright, \varepsilon, \dots, \triangleright, \varepsilon,) \overset{M^*}{\longmapsto} (q, u_1, w_1, \dots, u_k, w_k)$ mit $q \in \{h, h_y, h_n\}$ und $t = \sum_{i=2}^{k-1} (\mid u_i w_i \mid -1)$.
- M ist $f(n)$ *platzbeschränkt*, falls für jede Eingabe $x \in (\Sigma - \{\#, \triangleright\})^*$ mit $|x| = n$, $M(x)$ auf Platz $\leq f(n)$ berechnet wird.

Komplexitätsklassen

Anmerkungen:

$f(n)$ ist eine *Platzschranke* für M.

SPACE($f(n)$) ist die Klasse aller Sprachen $L \subseteq (\Sigma - \{\#, \triangleright\})^*$, die mit einer $f(n)$ platzbeschränkten Turingmaschine M (mit Ein- und Ausgabe) entschieden werden können.

Satz

Sei $L \in SPACE(f(n))$. Für jedes $\epsilon > 0$ ist $L \in SPACE(\epsilon \cdot f(n) + O(1))$.

Beweis.

Analog zum Speed-Up Theorem. □

Komplexitätsklassen

Definition

$$\mathrm{P} = \bigcup_{k \geq 1} \mathrm{TIME}(n^k)$$

$$\mathrm{PSPACE} = \bigcup_{k \geq 1} \mathrm{SPACE}(n^k)$$