Untere Schranken für den Platzbedarf bei der kontext-freien Analyse

Helmut Alt
Kurt Mehlhorn

Fachbereich 10 - Angewandte Mathematik und Informatik der Universität des Saarlandes

D-6600 Saarbrücken
Im Stadtwald

A 75/13
September 1975

1. Einleitung

Die kontext-freien Sprachen sind ein wichtiges Gebiet sowohl der theoretischen als auch der praktischen Informatik. Besondere Aufmerksamkeit wurde der Komplexität von Analysealgorithmen gewidmet.

Lewis, Hartmanis und Stearns [2] erzielten die ersten Resultate: Jede kontext-freie Sprache kann mit Speicheraufwand $O(\log^2 n)$ auf einer deterministischen Turingmaschine analysiert werden, es gibt kontext-freie Sprachen, die mit $O(\log \log n)$ erkannt werden können und $O(\log \log n)$ Speicherplatz ist zum Erkennen einer jeden nichtregulären Sprache nötig. Neuere Resultate von Sudborough [9] und Monien [8] lassen vermuten, daß die obere Schranke von $O(\log^2 n)$, wenn überhaupt, dann nur sehr schwer zu verbessern ist. Sie zeigten, daß aus der Existenz eines allgemeinen deterministischen Analysealgorithmus der Speicherkomplexität $O(\log n)$ die Gleichheit der deterministischen und nichtdeterministischen kontextsensitiven Sprachen folgt, also eine Lösung für Myhill's berühmtes lba-Problem.

Allerdings ist es denkbar, daß große Teilklassen der kontextfreien Sprachen, z.B. die deterministischen Sprachen, mit logarithmischem Platzaufwand erkennbar sind. Erste Resultate in dieser Richtung wurden von Hotz und Messerschmidt [5] und Mehlhorn [6] erzielt. Sie zeigen, daß Dycksprachen bzw. Klammersprachen Bandkomplexität $\log n$ haben.

In dieser Arbeit befassen wir uns mit unteren Schranken für die Bandkomplexität von deterministischen Sprachen . Die von Lewis, Hartmanis und Stearns angegebene kontext-freie Sprache der Bandkomplexität $O(\log \log n)$ ist nämlich inherent nichtdeterministisch. Wir erhalten folgende Resultate.

Definition: Eine kontext-freie Sprache L heißt **stark nicht-regulär**, falls es Worte u,v,w,x,y gibt, so daß $L \cap uv^* wx^* y$ nicht regulär ist.

Hauptsatz: Das Wortproblem einer jeden stark nicht-regulären kontext-freien Sprache benötigt O(log n) Speicherplatz unendlich oft.

Für einige Familien von eindeutigen kontext-freien Sprachen können wir zeigen, daß jede nicht-reguläre Sprache sogar stark nicht-regulär ist. So erhalten wir:

Korollar: Sei $L \subseteq a^* b^*$ kontext-frei aber nicht regulär. Dann hat das Wortproblem für L die Bandkomplexität O(log n).

In diesem Fall ist log n sowohl eine untere als auch obere Schranke für die Bandkomplexität. Die obere Schranke folgt aus dem Satz von Parikh. Für die Klasse der strikten Realzeitsprachen (strict realtime languages von Harrison und Havel [3]) erhalten wir nur eine untere Schranke.

Korollar: Sei L eine nicht-reguläre strikte Realzeitsprache. Dann hat das Wortproblem für L mindestens die Bandkomplexität O(log n).

Diese Resultate weisen auf einen Antagonismus im Konzept "Einfachheit" hin. Einfache Erzeugungsalgorithmen (strikt Realzeit) können keine ganz einfachen (Bandkomplexität O(log log n)) Sprachen erzeugen. Dies hat folgenden Grund: Die Funktion log n ist die am schwächsten wachsende band-konstruierbare Funktion. Sprachen geringerer Bandkomplexität müssen also die erkennende Maschine bei der Benutzung des Bandes leiten, um zu hohen Verbrauch zu vermeiden. Diese Steuerfunktion verlangt eine Ausdruckskraft, die von den einfachsten Generierungssystemen nicht geleistet werden kann.

Wir möchten den Herren Prof. G. Hotz und J. Messerschmidt für viele nützliche Diskussionen danken.

2. Vorbetrachtungen

Wenn wir im folgenden von 'Turingmaschine' reden, meinen wir damit eine deterministische Turingmaschine mit einem Eingabeband, auf dem sie nur lesen darf,und beliebigem Speicher (z.B. eine k-Band Turingmaschine, eine TM mit mehrdimensionalem Speicher, eine RAM oder einen Ballonautomaten). Sei $L : \mathbb{N} \to \mathbb{N}$ eine Abbildung. Wir sagen, eine Turingmaschine hat die Bandkomplexität L, falls bei keiner Eingabe der Länge n die Anzahl der im Speicher benutzten Zellen L(n) übersteigt.

Sei I das Eingabealphabet einer Turingmaschine M. Wir sagen, M erkennt $T \subset I^*$, wenn M zu jeder Eingabe $w \in I^*$ entscheidet,ob $w \in T$ oder nicht.
Es sei $L : \mathbb{N} \to \mathbb{N}$ eine Abbildung. Wir nennen eine formale Sprache T L-erkennbar, wenn es eine Turingmaschine der Bandkomplexität L(n) gibt, die T erkennt.
Wir beschränken uns zunächst auf kontext-freie nicht-reguläre Teilmengen von $\{a\}^* \cdot \{b\}^*$, wobei $\{a,b\}$ ein zweielementiges Alphabet ist, und zeigen , daß man zum Erkennen einer solchen Sprache mindestens O(log n) Speicherplatz benötigt.
Es gilt also der folgende Satz:

Satz: Es sei $\{a,b\}$ ein zweielementiges Alphabet. $L: \mathbb{N} \to \mathbb{N}$ sei eine Abbildung mit
$\lim\limits_{n\to\infty} \sup \frac{L(n)}{\log n} = 0$. Dann ist jede L-erkennbare kontext-freie Teilmenge von $\{a\}^* \cdot \{b\}^*$ regulär.

Ferner ist jede kontext-freie Teilmenge von $\{a\}^* \{b\}^*$ in O(log n) Platz erkennbar.

Eine Verallgemeinerung des Satzes ist das folgende Korollar:

Korollar: Es sei Σ ein Alphabet, $L \subset \Sigma^*$ sei kontext-frei und es gebe Wörter $u,v,w,x,y \in \Sigma^*$ mit : $T := \{u\}\cdot\{v\}^*\cdot\{w\}\cdot\{x\}^*\cdot\{y\} \cap L$ ist nicht regulär. Dann benötigt man zum Erkennen von T mindestens O(log n) Speicherplatz.

Beweis: Es ist klar, daß T als Durchschnitt einer kontext-freien mit einer regulären Sprache wieder kontext-frei ist. Damit ist auch $T' = \{a^i b^j \mid uv^i wx^j y \in T\}$ kontext-frei und nicht regulär. Zum Erkennen von T' braucht man also nach unserem Satz mindestens O(log n) Speicherplatz. Es sei nun M eine Turingmaschine,die T erkennt. Nun konstruieren wir eine Turingmaschine M' für T':

Zunächst überprüft M', ob die Eingabe aus $\{a\}^*\cdot\{b\}^*$ ist. Da dies eine reguläre Menge ist, wird dafür kein Speicherplatz benötigt.Liest nun M' das erste a, so verhält sie sich wie M bei der Eingabe von u und bleibt auf a stehen. Für jedes weitere a simuliert M' M bei der Eingabe von v. Beim ersten b verhält sich M' wie M bei der Eingabe von w, bleibt auf b stehen, dann wird bei jedem b M bei der Eingabe von x simuliert. Am Ende des Wortes verhält sich M' wie M bei der Eingabe von y. Es ist klar, daß die Simulation von M in der endlichen Kontrolle 'gesteuert' werden kann und daher keinen weiteren Speicherplatz verbraucht. M hat also den gleichen Speicherbedarf wie M' und damit auch mindestens O(log n).

3. Definition, Hilfssätze

Wir führen den Beweis des Satzes über Lemmata, die bis auf das erste Aussagen über gewisse Teilmengen von $\mathbb{N}_0^2$ sind.

Da wir im folgenden nur formale Sprachen betrachten, die Teilmengen von $\{a\}^* \cdot \{b\}^*$ sind, identifizieren wir eine solche Sprache L mit $\{(n,m) \mid a^n b^m \in L\} \subset \mathbb{N}_0^2$. Diese Zuordnung ist offensichtlich eindeutig. Wir beschäftigen uns also mit Teilmengen von $\mathbb{N}_0^2$ und übertragen die Begriffe "formale Sprachen", "kontext-frei", "regulär" usw. auf diese Teilmengen.

Zunächst benötigen wir einige Definitionen:

<u>Def. 1:</u> Seien $\alpha, \beta_1, \ldots, \beta_k \in \mathbb{N}_0^n$ $(n \in \mathbb{N})$ endlich viele Vektoren. Dann wird $L(\alpha; \beta_1, \ldots, \beta_k) \subset \mathbb{N}_0^n$ definiert durch

$$L(\alpha; \beta_1, \ldots, \beta_k) = \{\alpha + n_1\beta_1 + \ldots + n_k\beta_k \mid n_1, \ldots, n_k \in \mathbb{N}_0\}$$

Teilmengen von $\mathbb{N}_0^n$ dieser Art heißen <u>lineare Mengen</u>, endliche Vereinigungen davon <u>halblineare Mengen</u>.

Wir interessieren uns im folgenden nur für den Fall $n = 2$.

<u>Def. 2:</u> Eine lineare Teilmenge $L(\alpha; \beta_1, \ldots, \beta_k) \subset \mathbb{N}_0^2$ heißt <u>x-(y-)regulär</u>, falls es ein $r \in \mathbb{N}$ gibt mit $(r,o) \in \{\beta_1, \ldots, \beta_k\}$ $((o,r) \in \{\beta_1, \ldots, \beta_k\})$.

Eine lineare Menge der Form $L(\alpha; (r,o), (o,s))$ $\alpha \in \mathbb{N}_0^2$; $r, s \in \mathbb{N}_0$ heißt <u>Gitter</u> und wird auch mit $G(\alpha; r,s)$ bezeichnet.

<u>Bemerkung:</u> Da ein Gitter $G((\alpha_1, \alpha_2); r,s)$ offenbar der Sprache $\{a^{\alpha_1}\} \cdot \{a^r\}^* \cdot \{b^{\alpha_2}\} \cdot \{b^s\}^*$ entspricht, ist jedes Gitter regulär.

<u>Def. 3:</u> Eine Teilmenge $K \subset \mathbb{N}_0^2$ heißt <u>Kegel</u>, falls es $c_1, c_2 \in \mathbb{Q}$ und $m_1, m_2 \in \mathbb{Q}_+ \cup \{\infty\}$ gibt ($\mathbb{Q}_+ = \{x \in \mathbb{Q} \mid x \geq 0\}$) mit $m_1 \leq m_2$ und

$$K = \{(x,y) \in \mathbb{N}_0^2 \mid m_2 x + c_2 \geq y \geq m_1 x + c_1\}.$$

K heißt <u>entartet</u>, falls $m_1 = m_2$

<u>x-regulär</u>, falls $m_1 = 0$

<u>y-regulär</u>, falls $m_2 = \infty$

Die Geraden $g_i = \{(x,y) \in \mathbb{Q}_+^2 \mid y = m_i x + c_i\}$ $(i=1,2)$ heißen <u>Begrenzungsgeraden</u> von K.

Für $\alpha, \beta_1, \beta_2 \in \mathbb{N}_0^2$ bezeichnen wir mit $K(\alpha; \beta_1, \beta_2)$ den Kegel, dessen Begrenzungsgeraden durch α verlaufen und die die Richtung von β_1 bzw. β_2 haben.

Es sei T eine lineare Menge. Dann bezeichnen wir mit K(T) den kleinsten T enthaltenden Kegel.

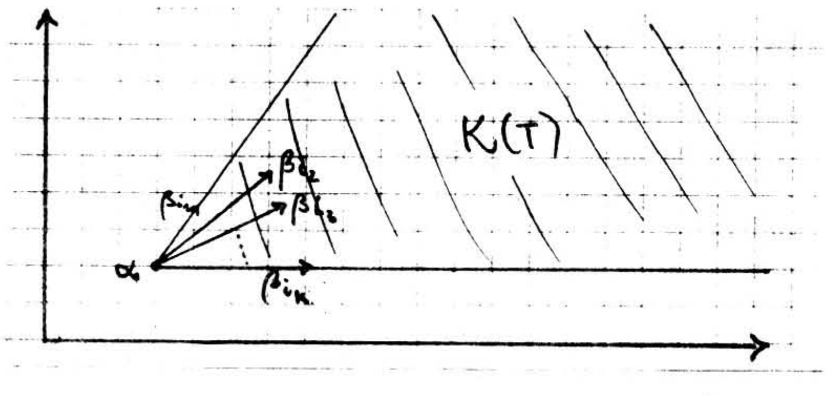
Wie man leicht sieht ist

$$K(L(\alpha; \beta_1, \ldots, \beta_k)) = K(\alpha; \beta_i, \beta_j) \text{ wobei } \beta_i$$

der Vektor aus $\{\beta_1, \ldots, \beta_k\}$ mit der stärksten und β_j der mit der schwächsten Steigung ist.

Es ist also auch K(T) genau dann x-(y-)regulär, wenn dies für T gilt.

$T = L(\alpha; \beta_1, \ldots, \beta_k)$
(x-regulär)



Die Beweisidee zu unserem Satz ist folgende:

Ist L eine kontex-freie Teilmenge von $\mathbb{N}_0^2$, so existieren nach dem Satz von Parikh lineare Mengen $T_1, \ldots, T_n$ mit $L = \bigcup_{i=1}^{n} T_i$.

Da wir zeigen, daß alle sowohl x- als auch y-regulären linearen Mengen regulär sind, können wir diese o.w.E. aus $T_1, \ldots, T_n$ ausschließen. Aus den restlichen konstruieren wir einen x-regulären Kegel K_1 und einen y-regulären K_2 mit:

Es gibt Gitter $G_{11},\dots,G_{1s_1}$, $G_{21},\dots,G_{2s_2}$

mit

$$L \cap K_1 = (G_{11} \cup \dots \cup G_{1s_1}) \cap K_1$$

und
$$L \cap K_2 = (G_{21} \cup \dots \cup G_{2s_2}) \cap K_2 \quad .$$

Dabei wird $K_1 (K_2)$ so gewählt, daß die Steigung seiner oberen (unteren) Begrenzungsgeraden $g_1 (g_2)$ höchstens die des Vektors aus der Vereinigung der 'Erzeugendensysteme' von $T_1,\dots,T_n$ ist, der die kleinste (größte) Steigung $\neq 0$ ($\neq \infty$) hat. Zu $K_1 \cap L (K_2 \cap L)$ liefern also nur noch die x-(y-)regulären Mengen aus $T_1,\dots,T_n$ einen Beitrag und für diese läßt sich zeigen, daß sie "im wesentlichen" nur aus einem Kegel geschnitten mit endlich vielen Gittern bestehen. Sei nun $G = G_{11} \cup \dots \cup G_{1s_1} \cup G_{21} \cup \dots \cup G_{2s_2}$.

G ist offenbar regulär. Aus der Nichtregularität von L folgt die von $G \setminus L$ oder $L \setminus G$. Beide Mengen liegen 'im wesentlichen' zwischen den beiden Geraden g_1 und g_2 und aus einer Turingmaschine M, die L erkennt, läßt sich eine Turingmaschine M' mit gleichem Platzbedarf konstruieren, die $G \setminus L$ bzw. $L \setminus G$ erkennt. Ist der Platzbedarf von M und damit der von M' kleiner als O(log n), so liegt aber ein ganzes Gitter in der von M' erkannten Sprache, wie man aus Lemma 1 sieht. $G \setminus L$ und $L \setminus G$ können aber beide kein Gitter mehr enthalten, da beide keine Elemente mit K_1 oder K_2 gemeinsam haben. Somit muß der Platzbedarf von M mindestens O(log n) sein.

Wir beweisen zunächst ein Lemma über Sprachen $\subset \mathbb{N}_0^2$, die mit weniger als O(log n) Platzbedarf erkennbar sind.

<u>Lemma 1:</u> Es sei $T \subset \mathbb{N}_0^2$ und es existiere eine Turingmaschine M der Bandkomplexität L, die T (d.h. $\{a^n b^m \mid (n,m) \in T\}$) erkennt, wobei $\lim_{n\to\infty} \frac{L(n)}{\log n} = 0$.

Dann gibt es für alle $0 < c_1 \le c_2$ $(c_1, c_2 \in \mathbb{Q}_+)$ ein $N \in \mathbb{N}$, so daß für alle $(n,m) \in T$ mit $c_1 n \le m \le c_2 n$ und $n \ge N$ gilt:

$$(n + k_1 n!\ ,\ m + k_2 m!) \in T \quad \text{für alle } k_1, k_2 \in \mathbb{N}_0.$$

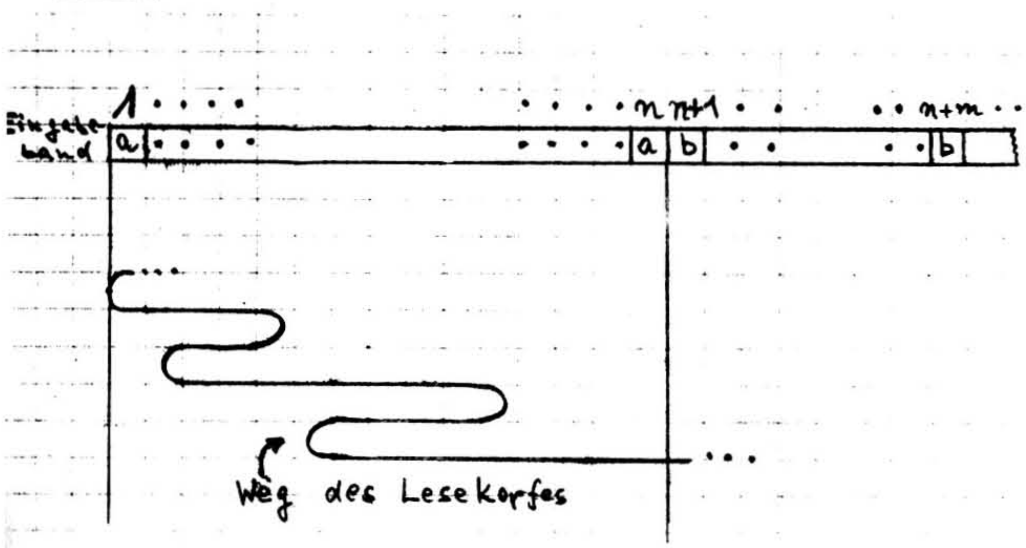
<u>Beweis:</u> Im folgenden wollen wir unter einem Maschinenzustand von M ein Paar aus $S \times K$ verstehen, wobei S die Menge der Zustände und K die der Speicherkonfigurationen ist. Eine Konfiguration von M ist also vollständig beschrieben durch Angabe des Maschinenzustands und der Stellung des Lesekopfes.

Die Maschine akzeptiere nun eine Eingabe $a^n b^m$.

<u>Beh. 1:</u> Ist die Anzahl der Maschinenzustände, die M nach der Eingabe von $a^n b^m$ annimmt kleiner als $\min(n,m)$, so wird auch $a^{n+k_1 n!}\ b^{m+k_2 m!}$ für alle $k_1, k_2 \in \mathbb{N}_0$ akzeptiert.

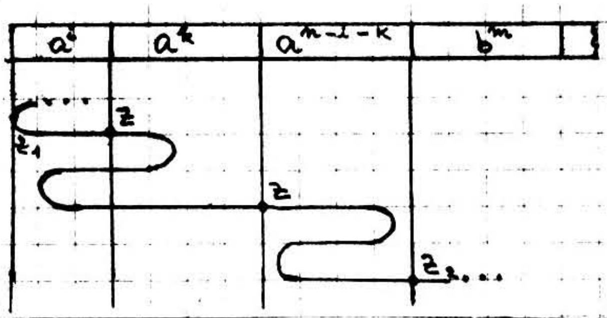
Beweis (informal): Wir geben der Maschine $a^n b^m$ ein. Mit a-Lauf bezeichnen wir einen Lauf des Lesekopfes zwischen Wortanfang und Grenze zwischen a's und b's (gleich in welcher Richtung), ohne daß eine Begrenzung (Wortanfang, Grenze a's - b's) mehrfach überlaufen wird.

<u>Bsp.:</u>

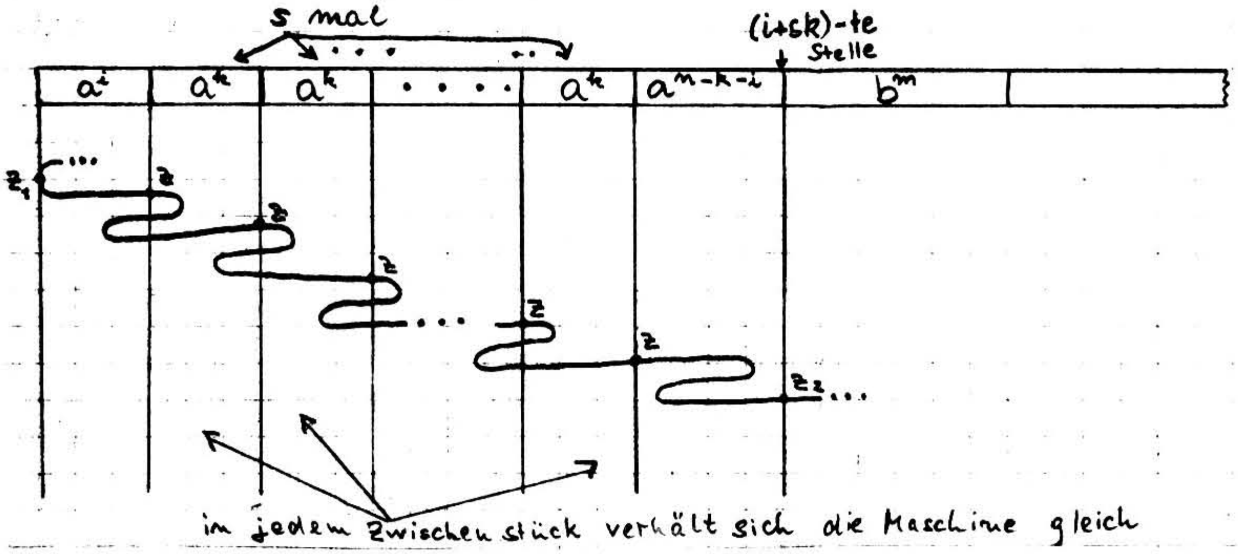


Beh. 1a: Bei Eingabe von $a^n b^m$ beginne ein a-Lauf mit dem Maschinenzustand z_1 und ende mit dem Maschinenzustand z_2. Dann führt die Maschine auch einen im Zustand z_2 endenden a-Lauf aus, wenn man den Lesekopf im Zustand z_1 am Anfang des Wortes $a^{n+k_1 n!} b^m$ beginnen läßt ($k_1 \in \mathbb{N}_0$ beliebig).

Beweis: Da es mehr a's als Maschinenzustände gibt, wird bei einem a-Lauf mindestens ein Zustand z an verschiedenen Stellen angenommen, etwa an der i-ten und (i+k)-ten Stelle.



Wir überlegen uns nun, daß für jedes $s \in \mathbb{N}_0$ die Maschine einen im Zustand z_2 endenden a-Lauf ausführt, wenn man den Lesekopf an den Anfang des Wortes $a^{n+sk} b^m$ setzt, während die Maschine sich im Zustand z_1 befindet.



Von der (i+sk)-ten Stelle ist nun der Lauf des Lesekopfes bis zur Grenze zwischen den a's und b's wieder der gleiche und es werden auch dieselben Zustände von der Maschine angenommen. Damit wird die Grenze zwischen a's und b's auch im Zustand z_2 erreicht.

Mit crossing sequence zwischen zwei Stellen i,i+1 des Lesebandes wollen wir, ähnlich wie in [4] , die Folge der Maschinenzustände, in die die Maschine beim Überschreiten dieser Grenze jeweils übergeht. Nur betrachten wir hier nicht nur Zustände der endlichen Kontrolle, sondern Paare aus S×K.

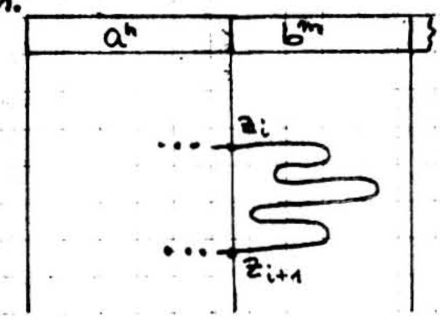
Beh. 1b: M erzeugt bei $a^{n+k_1 n!} b^m$ an der Grenze zwischen a's und b's für alle $k_1 \in \mathbb{N}_0$ die gleiche crossing sequence.

Beweis: Wir zeigen zunächst, daß die ersten Glieder der crossing sequences übereinstimmen.
Die Grenze zwischen a's und b's wird am Ende des ersten a-Laufs zum ersten Mal erreicht. (Es ist klar, daß sie überhaupt einmal erreicht wird, denn sonst würden mit $a^n b^m$ auch alle Worte aus $a^n \cdot \{a,b\}^*$ akzeptiert.)
Vor Beginn des ersten a-Laufs verhält sich die Maschine offenbar bei allen Eingaben gleich, da der Lesekopf noch nicht über eine gewisse Stelle $k \leq n$ hinausgekommen ist. Zu Beginn des ersten a-Laufs ist die Maschine also bei allen Eingaben obiger Form im gleichen Zustand und aus Beh. 1a folgt, daß sie es dann auch am Ende dieses a-Laufs ist.

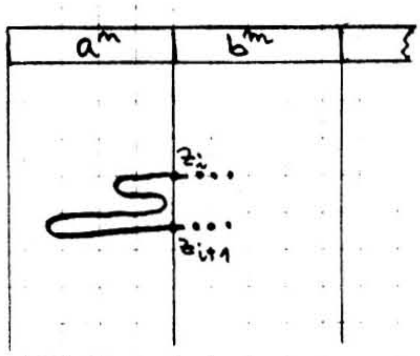
Als nächstes zeigen wir:

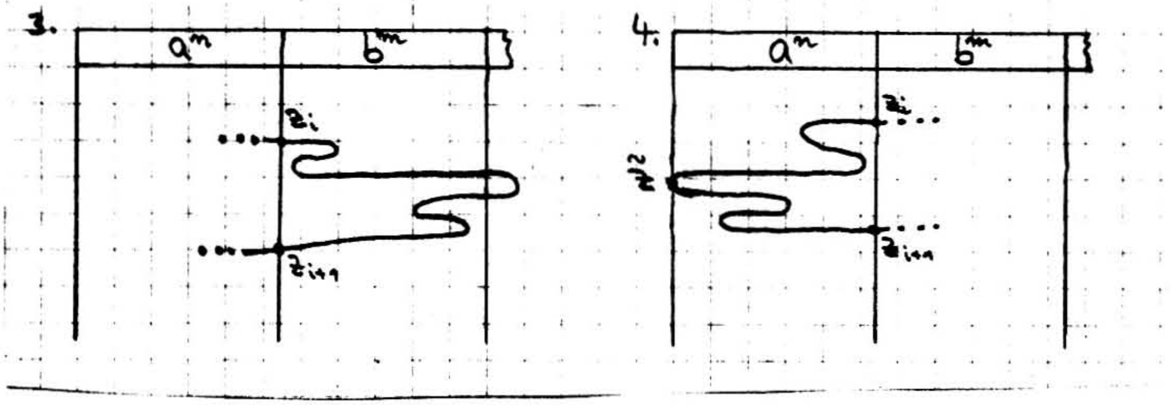
Stimmen bei allen Eingaben die i-ten Glieder der betrachteten crossing sequences überein, so stimmen auch die (i+1)-ten überein.
Es können bei Eingabe von $a^n b^m$ die folgenden Fälle auftreten:

1.



2.





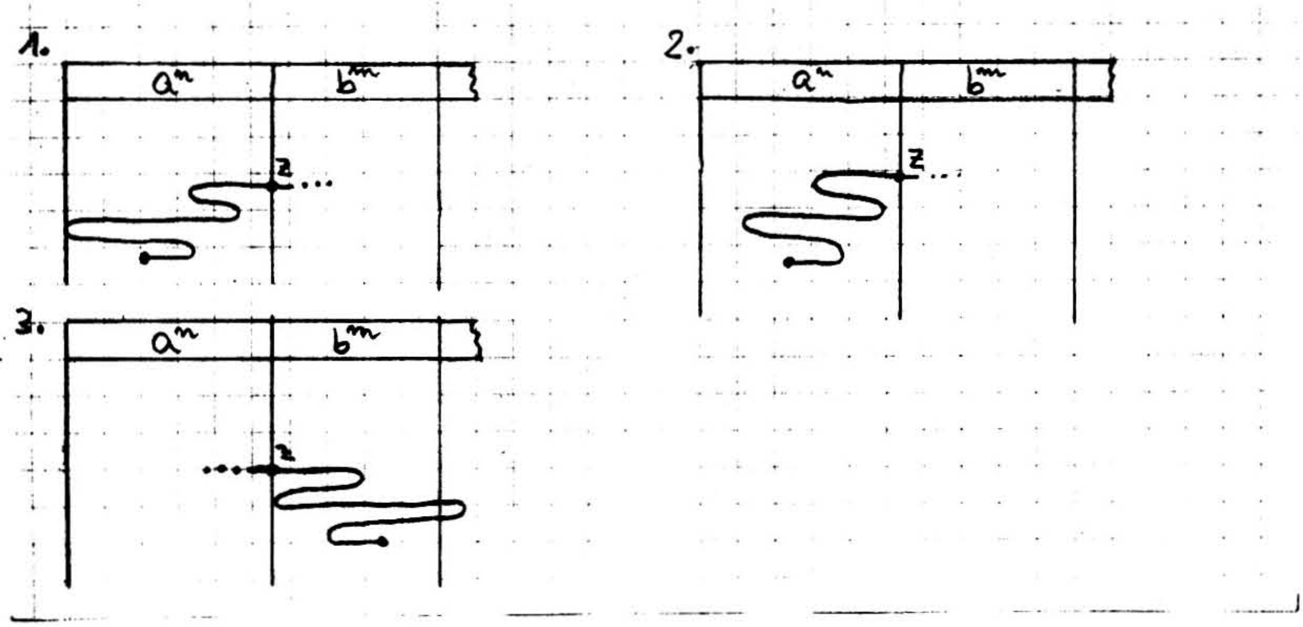
In den Fällen 1 und 2 verhält sich M offensichtlich, da das Wortende nicht erreicht wird, bei allen Eingaben $a^{n+k_1 n!} b^m$ gleich.

Im Fall 3 ebenfalls, da nur b's gelesen werden, deren Anzahl bei beiden Eingaben gleich ist. Im Fall 4 findet bei Eingabe von $a^{n+k_1 n!} b^m (K_1 \in \mathbb{N})$ nach Beh. 1a ein a-Lauf statt und am Anfang des Wortes wird ebenfalls der Zustand $\tilde{z}$ erreicht. Bei dem darauf folgenden mit $\tilde{z}$ beginnenden a-Lauf in umgekehrter Richtung wird ebenfalls z_{i+1} erreicht (Beh. 1a).

Damit stimmen also die crossing sequences bei allen Eingaben $a^{n+k_1 n!} b^m$ überein.

Sei z der Maschinenzustand, bei dem der Lesekopf diese Grenze zum letzten Mal überschreitet (Eingabe: $a^n b^m$).

Drei Fälle sind nun möglich:



In den Fällen 2 und 3 folgt aus obigen Überlegungen, im Fall 1 aus Beh. 1a, daß die Maschine bei Eingabe von $a^{n+k_1 n!} b^m$ gleichen Endzustand hält und mit $a^n b^m$ auch $a^{n+k_1 n!} b^m$ akzeptiert für alle $k_1 \in \mathbb{N}_0$. Dabei werden ebensoviele Maschinenzustände angenommen wie bei Eingabe von $a^n b^m$ also auch weniger als m. Analog zum vorherigen zeigt man nun, daß für jedes $k_2 \in \mathbb{N}_0$ auch $a^{n+k_1 n!} b^{m+k_2 m!}$ akzeptiert wird.

Zum Beweis von Lemma 1 ist nun noch zu zeigen:

Beh. 2: Sind die Voraussetzungen von Lemma 1 erfüllt, so gibt es ein $N \in \mathbb{N}$ mit:

Für alle $n \geq N$ und $m \in \mathbb{N}$ mit $c_1 m \leq n \leq c_2 m$ nimmt die Maschine bei Eingabe von $a^n b^m$ weniger als $\min(n,m)$ verschiedene Maschinenzustände an.

Beweis: Da $\lim_{n \to \infty} \frac{L(n)}{\log n} = 0$ ist, gibt es zu jedem $k > 0$ ein $K \in \mathbb{N}$ mit $k \cdot L(n) < \log n \quad \forall n \geq K$.

Nun gibt es bei einer Eingabe der Länge l offenbar höchstens $s \cdot L(l)^q \cdot r^{L(l)}$ verschiedene Maschinenzustände, wobei

s = Anzahl der inneren Zustände von M
q = Anzahl der Köpfe von M auf dem Arbeitsspeicher
r = Mächtigkeit des Alphabets von M.

Soll Beh. 2 nicht gelten, so muß es eine komponentenweise gegen ∞ strebende Folge $((n_i,m_i))_{i \in \mathbb{N}}$ auf $\mathbb{N}_0^2$ geben mit $c_1 m_j \leq n_j \leq c_2 m_j \quad \forall j \in \mathbb{N}$, und zu jedem $j \in \mathbb{N}$ einen $a^{n_j} b^{m_j}$ akzeptierenden Lauf von M, bei dem mehr als $\min(n_j,m_j)$ verschiedene Zustände angenommen werden.

Also

$$s \cdot L(l_j)^q \cdot r^{L(l_j)} \geq \min(n_j,m_j) \quad \text{mit } l_j = n_j + m_j \quad \forall j \in \mathbb{N}$$

Da $(1+c_2)\ n_j \geq l_j$ und $(1+\frac{1}{c_1})\ m_j \geq l_j$

$\Longrightarrow$ für $c = \min\left(\frac{1}{1+c_2}, \frac{1}{1+\frac{1}{c_1}}\right)$ gilt:

$$s \cdot L\ (l_j)^q \cdot r^{L(l_j)} \geq c \cdot l_j \quad \forall j \in \mathbb{N}$$

$\Longrightarrow \log s + q \cdot \log L(l_j) + L(l_j) \log r \geq \log c + \log l_j$

Da $\lim\limits_{j\to\infty} l_j = \infty$ muß auch gelten:

$\lim\limits_{\to\infty} L(l_j) = \infty$, wenn die Ungleichung für alle $j \in \mathbb{N}$ gelten soll.

$\Rightarrow$ es gibt also ein $c' > o$ mit

$$c' \cdot L(l_j) \geq \log l_j \quad \text{für hinreichend große } j \in \mathbb{N}$$

und somit ist

$$\limsup_{j\to\infty} \frac{L(l_j)}{\log l_j} \geq \frac{1}{c'} > o$$

d.h. es gibt eine nicht gegen o konvergierende Teilfolge von

$\left(\frac{L(i)}{\log i}\right)_{i \in \mathbb{N}}$ im Widerspruch zur Voraussetzung.

Wir haben im Beweis von Lemma 1 eine deterministische Turingmaschine verwendet. Es geht aber aus dem Beweis hervor, daß diese Voraussetzung nicht erforderlich ist. Gibt es nämlich einen $a^n b^m$ akzeptierenden Lauf einer nichtdeterministischen Turingmaschine, für die die Voraussetzungen von Lemma 1 gelten, so gibt es auch einen $a^{n+k_1 n!}\, b^{m+k_2 m!}$ akzeptierenden Lauf für alle $k_1, k_2 \in \mathbb{N}$.

Wir sehen außerdem am Beweis, daß weder die Art des Speichers noch die des Zugriffs auf den Speicher von Bedeutung sind, sondern lediglich die Anzahl der möglichen Speicherzustände. Das Lemma 1 und damit der Satz gelten daher für Maschinen mit beliebigem Speicher, wie z.B. k-Band-Turingmaschinen, Turingmaschine mit mehrdimensionalem Speicher, Pushdown-Automaten (auch 2-Weg) oder Random-Access-Maschinen (dabei wird Speicherbedarf definiert als max. Anzahl der benutzten Register · log (größte in einem Register vorkommende Zahl)).

Lemma 1 ist für den Beweis des Satzes sehr wesentlich. Mindest-Platzbedarf O(log n) sogar für nichtdeterministische Turingmaschinen zum Erkennen linearer nichtregulärer Teilmengen von $\mathbb{N}_0^2$ läßt sich schon aus Lemma 1 folgern. Also z.B. für

$\{a^n b^m \mid n \in \mathbb{N}\}$, $\{a^n b^m \mid m < n\}$ oder

$\{a^n b^m \mid c_1 n \leq m \leq c_2 n\}$.

Zum Beweis des Satzes für beliebige kontext-freie nichtreguläre Teilmengen von $\mathbb{N}_0^2$ benötigen wir noch einige Hilfssätze über lineare Teilmengen von $\mathbb{N}_0^2$:

Lemma 2: Für eine x-(y-)reguläre nichtentartete lineare Menge $T \subset \mathbb{N}_0^2$ gilt:

Es gibt einen x-(y-)regulären nichtentarteten Kegel K und Gitter $G_1, \ldots, G_p$ mit $K \cap T = K \cap (G_1 \cup \ldots \cup G_p)$.

Beweis: (für x-reguläres T).

Sei $T = L(\alpha; \beta_1, \ldots, \beta_n)$. $\beta_1 = (b_1, b_2)$ sei der erzeugende Vektor mit der größten Steigung. $\beta_2 = (b_1', 0)$ sei einer mit Steigung 0.

Wir zeigen zunächst:

Beh. 1: Es gibt ein Gitter G mit $T \cap G = K(T) \cap G$.

Beweis: Wir wählen $G = G(\alpha; b_1 b_1', b_2 b_1')$

falls $b_1 \neq 0$ ist, andernfalls $G = (\alpha; b_1', b_2)$.

Zu zeigen ist: $K(T) \cap G \subset T$

Sei also $\delta = (d_1, d_2) \in K(T) \cap G$.

$\Longrightarrow$ 1) $\exists\, n_1, n_2 \in \mathbb{N}_0$ mit $\delta = \alpha + n_1 \gamma_1 + n_2 \gamma_2$

wobei $\gamma_1 = (b_1 \cdot b_1', 0)$, $\gamma_2 = (0, b_2 b_1')$

2) $d_2 \geq a_2$ und $\dfrac{d_2 - a_2}{d_1 - a_1} \leq \dfrac{b_2}{b_1}$

wobei $\alpha = (a_1, a_2)$

Aus 1) folgt: $d_1 - a_1 = n_1 b_1 b_1'$

$d_2 - a_2 = n_2 b_2 b_1'$

Aus 2) folgt somit:

$$\frac{n_2 b_2 b_1'}{n_1 b_1 b_1'} \leq \frac{b_2}{b_1}$$

also $n_1 \geq n_2$

Wie man leicht sieht, ist

$$\delta = \alpha + n_2 b_1' \beta_1 + (n_1 - n_2)\, b_1 \beta_2$$

und da $n_2 b_1' \in \mathbb{N}_0$ und nach dem vorherigen $(n_1 - n_2) b_1 \in \mathbb{N}_0$,

ist $\delta \in T$.

Wir wollen im folgenden einen Vektor $\delta_1 \in \mathbb{N}_0^2$ von $\delta_2 \in \mathbb{N}_0^2$ aus <u>erreichbar</u> nennen, wenn es $n_1, n_2 \in \mathbb{N}_0$ gibt mit $\delta_2 = \delta_1 + n_1 \beta_1 + n_2 \beta_2$, also wenn $\delta_2 \in L(\delta_1; \beta_1, \beta_2)$:

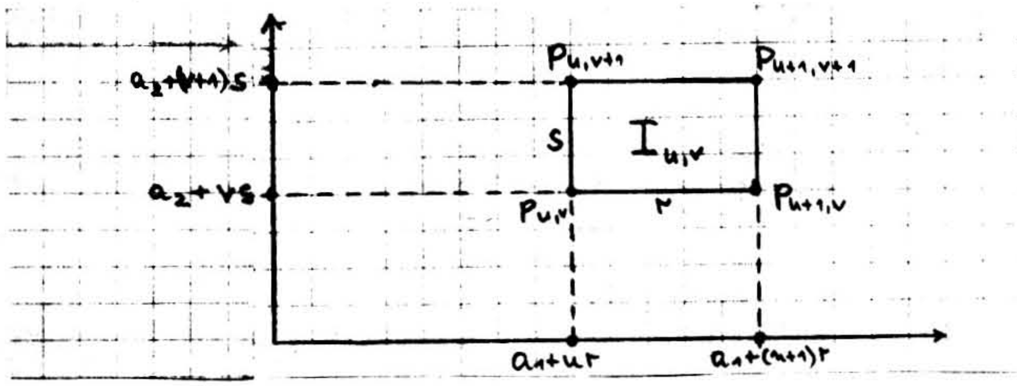
Sei nun $r := b_1b_1'$ $s := b_2b_1'$

Für $u,v \in \mathbb{N}_0$ sei

$$p_{u,v} := (a_1 + ur,\ a_2 + vs)$$

$$(\text{Also } p_{u,v} \in G \quad \forall u,v \in \mathbb{N}_0)$$

Sei $I_{u,v}$ der Inhalt der "Masche" von G mit der linken unteren Ecke $p_{u,v}$



Also $I_{u,v} = [a_1+ur,\ a_1+(u+1)r] \times [a_2+vs,\ a_2+(v+1)s]$

Es sei $I'_{u,v} = (I_{u,v} \cap T) - p_{u,v} \subset [o:r] \times [o:s]$

Man überlegt sich leicht:

<u>Beh. 2:</u> Ist $p_{u',v'}$ von $p_{u,v}$ aus erreichbar, so ist

$I'_{u,v} \subset I'_{u',v'}$

Es gibt nun ein $(u_o,v_o) \in \mathbb{N}_0^2$ mit

$$I'_{u_o,v_o} = I'_{u,v} \quad \text{für alle} \quad u,v \in \mathbb{N}_0 \text{ mit:}$$

$$p_{u,v} \text{ ist von } p_{u_o,v_o} \text{ aus erreichbar.}$$

Denn gäbe es ein solches (u_0,v_0) nicht, so könnte man durch Induktion mit Hilfe von Beh. 2 eine unendliche Kette

$I'_{u_1,v_1} \subsetneqq I'_{u_2,v_2} \subsetneqq \dots$ finden.

Da aber $I'_{u,v} \subset [0:r] \times [0:s] \quad \forall u,v \in \mathbb{N}_0$

also Teilmengen einer endlichen Menge sind, ist dies nicht möglich.

Seien also u_0, v_0 mit obiger Eigenschaft gewählt.

Wir setzen nun $K = K(p_{u_0,v_0}; \beta_1, \beta_2)$.

<u>Beh. 3:</u> Sei $x \in K$. Dann gibt es $u,v \in \mathbb{N}_0$ mit $x \in I_{u,v}$ und $p_{u,v} \in K$.

<u>Beweis:</u> Sei $x = (x_1, x_2)$.

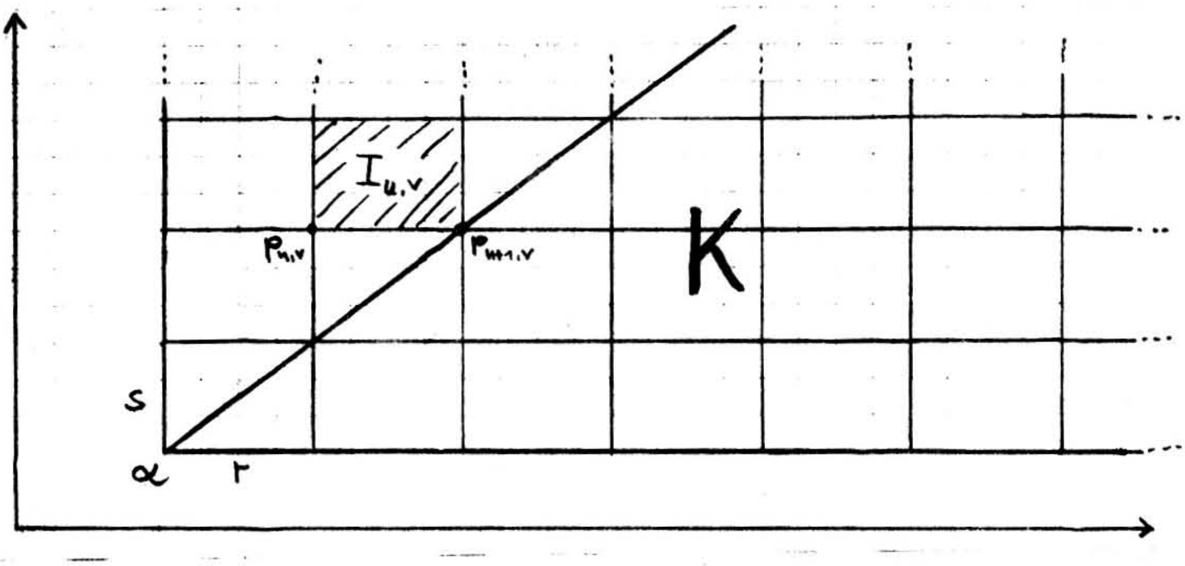
Zunächst überlegt man sich leicht, daß

$$x \in I_{u',v'} \quad \text{mit } u' = \left[\frac{x_1 - a_1}{r}\right] \qquad v' = \left[\frac{x_2 - a_2}{s}\right].$$

Dann zeigen wir:

Ist für ein $u,v \in \mathbb{N}_0$ $\;p_{u,v} \in \complement K$, (=Komplement von K),

so ist $I_{u,v} \setminus \{p_{u+1,v}\} \subset \complement K$



$p_{u,v} \in \complement K \Longrightarrow$ da $a_2 + vs \geqslant a_2$, gilt $\frac{vs}{ur} > \frac{s}{r}$

d.h. $\quad v > u$

$\Longrightarrow v + 1 > u$

$v + 1 > u + 1$

$v \geq u + 1$

d.h. von den vier Ecken von $I_{u,v}$ kann höchstens die linke untere auf dem Rand von K liegen.

Man überlegt sich leicht, daß dann

$$I_{u,v} \setminus \{ p_{u+1,v} \} \subset \complement K$$

Liegt also x in $K \cap I_{u,v}$ mit $p_{u,v} \notin K$

so ist $\quad x = p_{u+1,v} \in K$ also $x \in I_{u+1,v}$.

Wir kommen nun zum eigentlichen Beweis von Lemma 2:

Sei $I'_{u_o,v_o} = \{ x_1,\dots,x_p \}$

Wir setzen für $i = 1,\dots,p$:

$$G_i := G(p_{u_o,v_o} + x_i;\ r,s)$$

<u>Beh. 4:</u> $T \cap K = (G_1 \cup \dots \cup G_p) \cap K$.

<u>Beweis:</u>

$\subseteq$: Sei $x \in T \cap K$. Dann gibt es nach Beh. 3 ein $(u,v) \in \mathbb{N}_o^2$ mit $p_{u,v} \in K$ und $x \in I_{u,v}$. Nun ist aber jedes $p_{u,v} \in K$ von p_{u_o,v_o} aus erreichbar (da jedes $p_{u',v'} \in K(T)$ von α aus erreichbar ist (Beh. 1) und K nur eine Verschiebung von K(T) um den Vektor $(u_o r, v_o s)$ ist).

Nun ist aber $I'_{u,v} = I'_{u_o,v_o}$

$\Longrightarrow \exists\, i \in \{1,\dots,p\}$ mit

$x - p_{u,v} = x_i$

Also: $x = x_i + p_{u,v} = p_{u_o,v_o} + x_i \pm ((u-u_o)r,(v-v_o)s)$

also $x \in G_i$.

$\underline{\supset}$: Sei $x \in G_j \cap K$ für ein $j \in \{1,\ldots,p\}$

$\underset{\text{Beh.3}}{\Longrightarrow}$ $\exists u,v \in \mathbb{N}_o$ mit $x \in I_{u,v}$ und $p_{u,v}$ ist von p_{u_o,v_o} aus erreichbar

Also ist $x - p_{u,v} = x_j$

$\Longrightarrow$ $x - p_{u,v} + p_{u_o,v_o} = p_{u_o,v_o} + x_j \in T$ nach Def.

$\Longrightarrow$ $\exists n_1,\ldots,n_k \in \mathbb{N}_o$ mit

$$p_{u_o,v_o} + x_j = \alpha + n_1\beta_1 + \ldots + n_k\beta_k$$

Da $p_{u,v}$ von p_{u_o,v_o} aus erreichbar:

$\exists n_1', n_2'$ mit $p_{u,v} - p_{u_o,v_o} = n_1'\beta_1 + n_2'\beta_2$

Also $x = p_{u_o,v_o} + x_j + p_{u,v} - p_{u_o,v_o}$

$$= \alpha + (n_1+n_1')\beta_1 + (n_2+n_2')\beta_2 + n_3\beta_3 + \ldots + n_k\beta_k$$

Also $x \in T$.

Eine Verallgemeinerung von Lemma 2 ist

__Lemma 3:__ Sei $L \subset \mathbb{N}_o^2$ kontext-frei und nicht-regulär. Dann gibt es einen x-regulären nichtentarteten Kegel K_1 und einen y-regulären nichtentarteten Kegel K_2 und Gitter $G_{11},\ldots,G_{1s_1}, G_{21},\ldots,G_{2s_2}$ $(s_1,s_2 \in \mathbb{N})$ mit

$$L \cap K_i = (G_{i1} \cup \ldots \cup G_{is_i}) \cap K_i \qquad (i = 1,2).$$

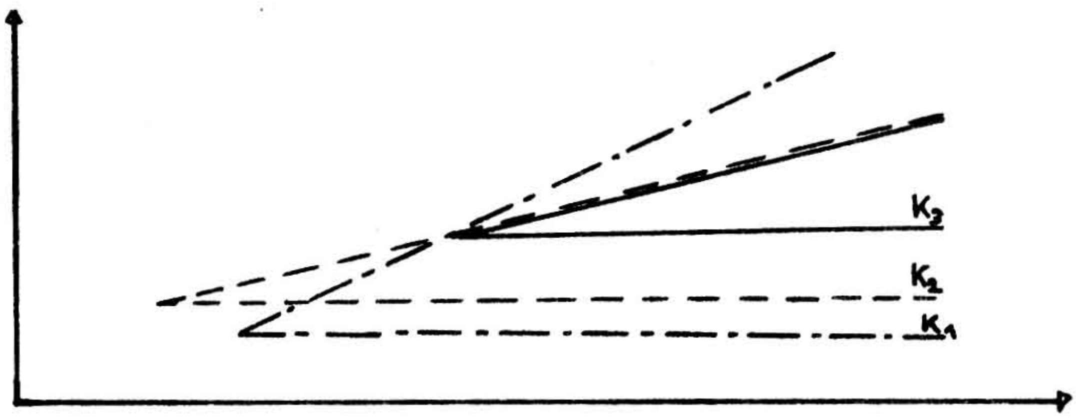
<u>**Beweis:**</u> (Wir konstruieren K_1; K_2 wird analog konstruiert)

Sei $L = T_1 \cup \ldots \cup T_n$ **die Zerlegung von L in lineare Teilmengen. (Nach dem Satz von Parikh läßt sich jede kontext-freie Teilmenge von $\mathbb{N}_0^2$ so zerlegen).**

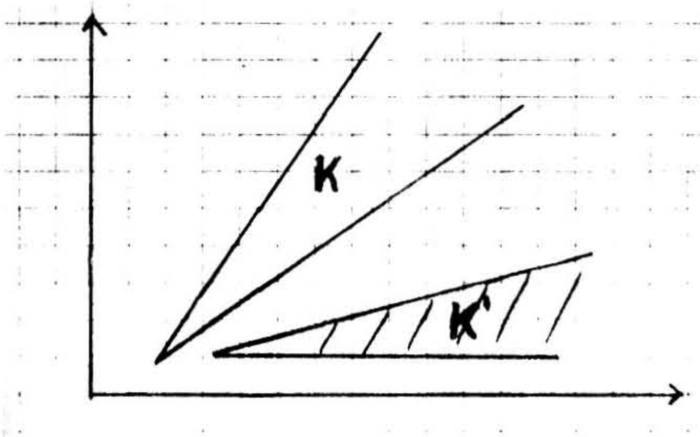
Dabei seien

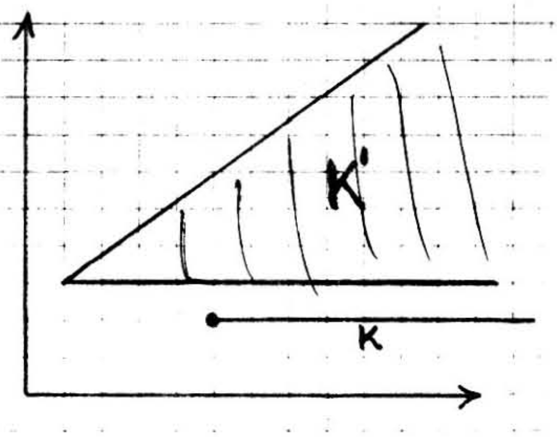
$T_1, \ldots, T_i$	$\begin{cases}\text{x-regulär}\\ \text{nichtentartet}\end{cases}$
$T_{i+1}, \ldots, T_j$	$\begin{cases}\text{y-regulär aber nicht}\\ \text{x-regulär}\end{cases}$
$T_{j+1}, \ldots, T_n$	$\begin{cases}\text{weder x- noch y-regulär}\\ \text{oder x-regulär und entartet}\end{cases}$

Offenbar enthält der Durchschnitt zweier nichtentarteter x-regulärer Kegel K_1 und K_2 wieder einen n.e. x-reg. Kegel K_3



Ebenso enthält das Komplement eines nicht x-regulären Kegels K oder das eines entarteten x-regulären Kegels einen nichtentarteten x-regulären Kegel K'.





Aus beidem folgt durch Induktion:
Der Durchschnitt endlich vieler Teilmengen von $\mathbb{N}_0^2$, die entweder n.e. x-reguläre Kegel oder Komplemente nicht x-regulärer Kegel sind, enthält einen nichtentarteten x-regulären Kegel.

Also auch:
(owE: $L \neq \emptyset$)

$$\mathcal{C}\left(\bigcup_{s=i+1}^{n} K(T_s)\right) \cap \bigcap_{r=1}^{i} \tilde{K}(T_r) = \bigcap_{s=i+1}^{n} \mathcal{C}K(T_s) \cap \bigcap_{i=1}^{i} \tilde{K}(T_r)$$

($\tilde{K}(T_r)$ sei dabei der nach Lemma 2 zu T_r konstruierte Kegel)

K_1 sei ein solcher darin enthaltener nichtentarteter x-regulärer Kegel.

Seien $G_1^{(r)} \ldots G_{j_r}^{(r)}$ die nach Lemma 1 zu $\tilde{K}(T_r)$ gehörigen Gitter $(r=1,\ldots,i)$.

Sei $s_1 = \sum_{r=1}^{i} j_r$ und

$$\begin{aligned} G_{11} &= G_1^{(1)} \\ &\vdots \\ G_{1s_1} &= G_{j_i}^{(i)} \end{aligned}$$

<u>Beh.:</u> $L \cap K_1 = (G_{11} \cup \ldots \cup G_{1s_1}) \cap K_1$

<u>$\subset$:</u> Sei $\alpha \in L \cap K_1$

nach Konstruktion existiert ein $p \in \{1,\ldots,i\}$ mit $x \in T_p$

Außerdem ist $x \in \tilde{K}(T_p)$

$\xRightarrow[\text{Lemma 2}]{} \exists\, 1 \leq r \leq j_p$ mit $x \in G_r^{(p)}$

$\underline{\supset:}$ Sei $x \in \bigcup_{i=1}^{s_1} G_{1i}$

$\Longrightarrow \exists p,\, q : q \leq j_p$ mit $x \in G_q^{(p)}$

außerdem : $x \in \tilde{K}(T_p)$

$\underset{\text{Lemma 2}}{\Longrightarrow}$ $x \in T_p \subset L.$

Analog zeigt man $L \cap K_2 = (G_{21} \cup \ldots \cup G_{2s_2}) \cap K_2$.
Wir zeigen nun noch:

<u>Lemma 4:</u> Es sei $L \subset \mathbb{N}_o^2$ kontext-frei und nicht-regulär. Dann gibt es eine komponentenweise gegen ∞ strebende Folge $((n_i, m_i))_{i \in \mathbb{N}}$ in L und $c_1, c_2 \in \mathbb{Q}_+$ mit $c_1 n_i \leq m_i \leq c_2 n_i \quad \forall i \in \mathbb{N}$.

<u>Beweis:</u> Sei $L = T_1 \cup \ldots \cup T_k$ eine Zerlegung von L nach dem Satz von Parikh in lineare Teilmengen. Da L nicht-regulär ist, existiert ein $i \in \{1, \ldots, k\}$ mit: T_i ist nicht-regulär. Dann können aber nicht alle erzeugenden Vektoren von T_i die Steigung o oder ∞ haben, denn sonst wäre T_i eine Vereinigung von Gittern. Sei also $\beta = (b_1, b_2)$ ein erzeugender Vektor von T_i mit $o < \frac{b_2}{b_1} < \infty$ und α sei der 'Eckpunkt' von T_i

Wir wählen

$$(n_i', m_i') = \alpha + i\beta \qquad \forall i \in \mathbb{N}$$

und

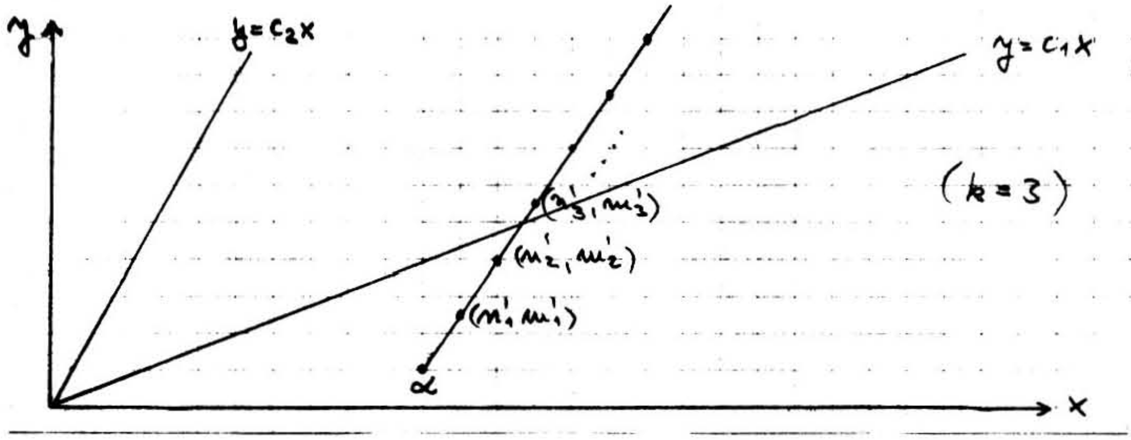
$$c_1, c_2 \in \mathbb{Q} \text{ bel. mit } o < c_1 < \frac{b_2}{b_1} < c_2 < \infty.$$

Man überlegt sich leicht, daß $((n_i', m_i'))_{i \in \mathbb{N}}$ komponentenweise gegen ∞ strebt und daß von einem bestimmten Index k an gilt:

$c_1 n_j' \leq m_j' \leq c_2 n_j'$.

Diese Teilfolge bezeichnen wir mit $((n_i, m_i))_{i \in \mathbb{N}}$.

Sie erfüllt offenbar die Beh. von Lemma 4.



3. Der Beweis des Satzes

Sei also nun $L \subset \mathbb{N}_0^2$ kontext-frei und nicht-regulär und wir nehmen an, L werde mit weniger als O(log n) Platzbedarf erkannt.

Seien K_1 und K_2, $G_{11},\dots,G_{1s_1},\dots,G_{1s_2}$ Kegel und Gitter mit den in Lemma 3 beschriebenen Eigenschaften. Es seien

$$G_1 := \bigcup_{i=1}^{s_1} G_{1i} \qquad G_2 := \bigcup_{j=1}^{s_2} G_{2j} \qquad G := G_1 \cup G_2$$

$$L_1 := G \setminus L \qquad L_2 := L \setminus G$$

Zunächst nehmen wir an,

L_2 sei regulär.

$\Longrightarrow$ L_1 ist nicht-regulär, da $L = (G \setminus L_1) \cup L_2$ nicht-regulär ist.

L_1 ist aber nach Korollar 5.6.2 S. 182 in [1] kontext-frei.

$\xRightarrow{\text{Lemma 4}}$ Es gibt $c_1, c_2 \in \mathbb{Q}_+ - \{0\}$ und eine Folge $((n_i, m_i))_{i \in \mathbb{N}}$ in L_1 mit den in Lemma 4 beschriebenen Eigenschaften.

Es seien für $i = 1,\ldots,s_1$ $\alpha_{1i}=(a_{1i},b_{1i})$, r_{1i},s_{1i} so gewählt,

daß $G_{1i} = (\alpha_{1i};r_{1i},s_{1i})$,

und für $j = 1,\ldots,s_2$ $\alpha_{2j}=(a_{2i},b_{2i})$, r_{2j},s_{2j} so, daß

$$G_{2j} = (\alpha_{2j};\ r_{2j},s_{2j}).$$

Wir wählen nun $k \in \mathbb{N}$ so groß, daß

1. $n_k \geq \max\limits_{i=1,\ldots,s_1} r_{1i}$
2. $m_k \geq \max\limits_{j=1,\ldots,s_2} s_{2j}$
3. $n_k \geq$ 1. Komponente der Punkte auf der linken Begrenzungsgeraden von K_2
4. $m_k \geq$ 2. Komponente der Punkte auf der rechten Begrenzungsgeraden von K_1
5. $n_k \geq N$, wobei N nach Lemma 1 die zu L_1 gehörige Konstante ist. Wird nämlich L mit weniger als $O(\log n)$ Platzbedarf erkannt, so gilt dies auch für das Komplement von L und auch für L_1 als Durchschnitt von $\complement L$ mit einer regulären Menge.

Da $L_1 \cap K_1 \cap G_1 \subset L_1 \cap L = \emptyset$

und $L_1 \cap K_2 \cap G_2 = \emptyset$, gibt es für (n_k,m_k) 3 Möglichkeiten:

1. $(n_k,m_k) \notin K_1 \cup K_2$
2. $(n_k,m_k) \in K_1 \cap \complement G_1$
3. $(n_k,m_k) \in K_2 \cap \complement G_2$

Außerdem existiert ein G_{js} mit $j \in \{1,2\}$, $s \in 1,\ldots,s_j$ und $(n_k,m_k) \in G_{js}$.

zu 1.) α) $j=1$

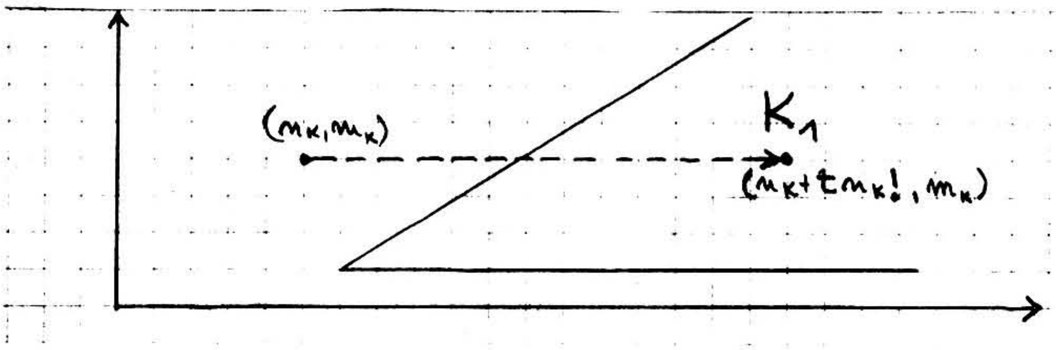
$\Longrightarrow \exists k \in \mathbb{N}$ mit $n_k = a_{1s} + kr_{1s}$

Außerdem ist $(n_k + t \cdot n_k!,\ m_k) \in L_1$ $\qquad \forall t \in \mathbb{N}_0$
(Lemma 1)

Da $n_k \geq r_{1s}$ nach Vor., gilt: $r_{1s} \mid n_k!$

also ist auch $(n_k + tn_k!,\ m_k) \in G_{1s}$ $\qquad \forall t \in \mathbb{N}_0$

Da m_k größer als die untere Begrenzung von K_1 ist, können wir durch hinreichend großes t erreichen, daß $(n_k + tn_k!,\ m_k) \in K_1$



Es gibt also ein $t \in \mathbb{N}$ mit:

$(n_k + tn_k!, m_k) \in L_1 \cap K_1 \cap G_{1s}$

$L_1 \cap K_1 \cap G_1 = \emptyset$

Widerspruch!

β) $j=2$: analoger Schluß mit der 2. Komponenten.

zu 2.) $(n_k, m_k) \in G_1 \Longrightarrow (n_k, m_k) \in G \setminus G_1 = G_2$
nun zu Fall 1.) analoger Schluß mit der 2. Komponenten

zu 3.) Wie 2.) mit 1. Komponente .
Wir erhalten also in jedem Fall einen Widerspruch, L_2 kann also nicht regulär sein, wenn die Annahme, daß L mit weniger als O(log n) Platzbedarf erkannt wird, richtig sein soll.

Es gibt nun in L_2 eine Folge $((n_i, m_i))_{i \in \mathbb{N}}$ und $c_1, c_2 \in \mathbb{Q}$ mit den Eigenschaften aus Lemma 4.

Wählen wir $k \in \mathbb{N}$ hinreichend groß, so läßt sich mit Lemma 1 schließen, daß das Gitter $\tilde{G} = ((n_k, m_k); n_k!, m_k!)$ Teilmenge von L_2 ist.

Es ist aber der Durchschnitt eines nichtentarteten Gitters mit einem nichtentarteten Kegel immer nichtleer.

Also $\quad L_2 \cap K_1 \supset \tilde{G} \cap K_1 \neq \emptyset$

Aber $\quad L_2 \cap K_1 \subset L \cap K_1 \subset G \quad$ nach Lemma 1

also $\quad L_2 \cap K_1 \subset L_2 \cap G = \emptyset$

Dies ist ein Widerspruch und die Annahme, L benötige weniger als O(log n) Speicherplatz, ist falsch, womit unser Satz bewiesen wäre.

5. Anwendung auf Familien von deterministischen Sprachen

In diesem Abschnitt wenden wir unsere Ergebnisse auf Familien von deterministischen Sprachen an. Die Notation wird von Valiant [10] übernommen.

Definition: R_1 ist die Menge, der in Realzeit mit leerem Speicher akzeptierbaren Mengen, d.h. $L \in R_1$, falls es einen deterministischen Kellerautomaten gibt, der keine ε-Züge macht und L mit leerem Speicher akzeptiert.

Harrison und Havel [3] untersuchten die Klassen der Realzeit und strikten Realzeitsprachen. Die Klasse R_1 liegt zwischen diesen beiden Familien.

Satz: Sei $L \in R_1$ nicht-regulär. Dann benötigt das Wortproblem für L unendlich oft logarithmischen Speicherplatz.

Beweis: Sei also $L \in R_1$ nicht-regulär. Wir werden zeigen, daß L dann sogar stark nicht-regulär ist.

Sei A ein deterministischer Kellerautomat, der L mit leerem Speicher akzeptiert und keine ε-Züge macht. Wir können o.B.d.A. annehmen, daß A bei jedem Übergang höchstens 1 Zeichen schreibt [10] . Sei $m = ||S|| \cdot ||\Gamma||$, wo S die Zustandsmenge von A und Γ das Kelleralphabet von A ist. Da L nicht-regulär ist, gibt es ein Wort $z \in L$, so daß die maximale Länge des Keller, die während der Rechnung von A auf z auftritt, m^2 übersteigt.

Wir können z schreiben als $z = z_1 z_2 z_3$ mit:

(1) $z_2 \neq \varepsilon$

(2) Der Keller von A ist leer, wenn A z_1 bzw. $z_1 z_2$ gelesen hat.

(3) Der Keller von A ist niemals leer, wenn A in z_2 liest und die Länge des Kellers erreicht $n > m^2$, während A in z_2 liest.

Für $o \leq i \leq n$ definieren wir:

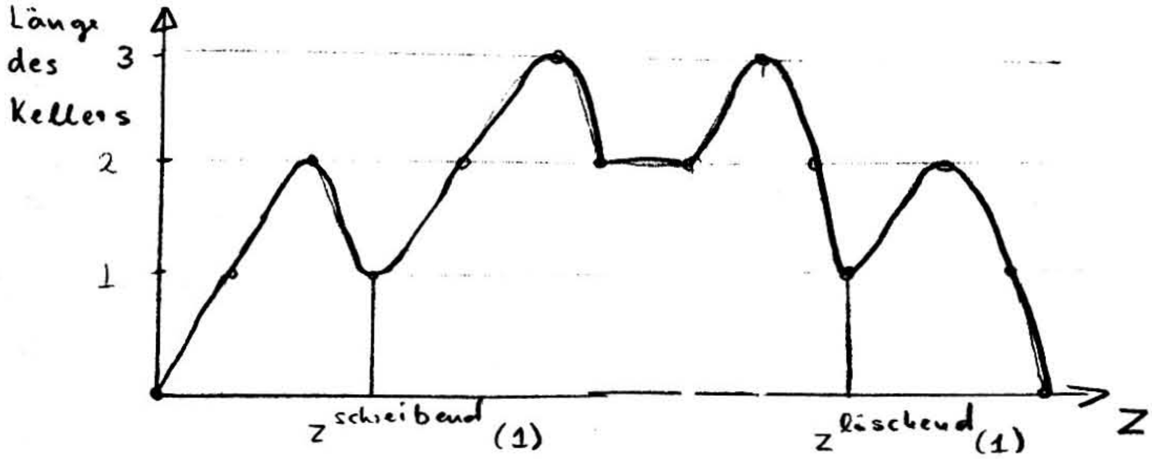
$z^{\text{schreibend}}$ (i) = der längste Präfix u von $z_1 z_2$ mit

1) nachdem A u eingelesen hat, beträgt die Länge des Kellers i
2) die Länge des Kellers erreicht n, bevor sie wieder i annimmt.

und

$z^{\text{löschend}}$ (i) = die kürzeste echte Verlängerung t von $z^{\text{schreibend}}$ (i) mit: nachdem A t gelesen hat, beträgt die Länge des Kellers i .

Beispiel



Sei nun $s^{\text{schreibend}}$ (i) ($\underset{\sim}{z}^{\text{schreibend}}$ (i)) der Zustand von A (das oberste Kellerzeichen), nachdem A $z^{\text{schreibend}}$ (i) gelesen hat. $s^{\text{löschend}}$ (i) und $\underset{\sim}{z}^{\text{löschend}}$ (i) sind analog definiert. Wegen $n > m^2$ gibt es i,j mit $i < j$, so daß gilt:

$$s^{schreibend}(i) = s^{schreibend}(j)$$
$$\underset{\sim}{z}^{schreibend}(i) = \underset{\sim}{z}^{schreibend}(j)$$
$$s^{löschend}(i) = s^{löschend}(j)$$
$$\underset{\sim}{z}^{löschend}(i) = \underset{\sim}{z}^{löschend}(j)$$

Wähle nun u,v,w,x,y so, daß gilt:

$$u = z^{schreibend}(i) ,$$
$$z^{schreibend}(j) = z^{schreibend}(i)\ v ,$$
$$z^{löschend}(j) = z^{schreibend}(j)\ w,$$
$$z^{löschend}(i) = z^{löschend}(j)\ x \quad \text{und}$$
$$z = z^{löschend}(i)\ y$$

Wir betrachten A auf den Wörtern der Gestalt uv^pwx^qy.

Sei $L' = uv^*wx^*y \cap L$.

<u>Beobachtung 1:</u> Für jedes $p \in \mathbb{N}$ gilt : $uv^pwx^py \in L$.

<u>Beobachtung 2:</u> Für jedes $q \in \mathbb{N}$ existiert ein $p_o \in \mathbb{N}$, so daß für alle $p \geq p_o$ gilt: $uv^pwx^qy \notin L$.

<u>Beweis:</u> Sei $p > q$. und setze A auf uv^pwx^q an. Nachdem A uv^pwx^q gelesen hat, beträgt die Länge des Kellers $i + (j - i) \cdot (p - q)$. Da A keine ε-Züge vornimmt, kann es die Länge des Kellers durch Lesen von y höchstens um $|y|$ Symbole verkürzen. Also ist $uv^pwx^qy \notin L$, falls $i + (j - i) \cdot (p - q) > |y|$.

<u>Beobachtung 3:</u> L' ist nicht-regulär.

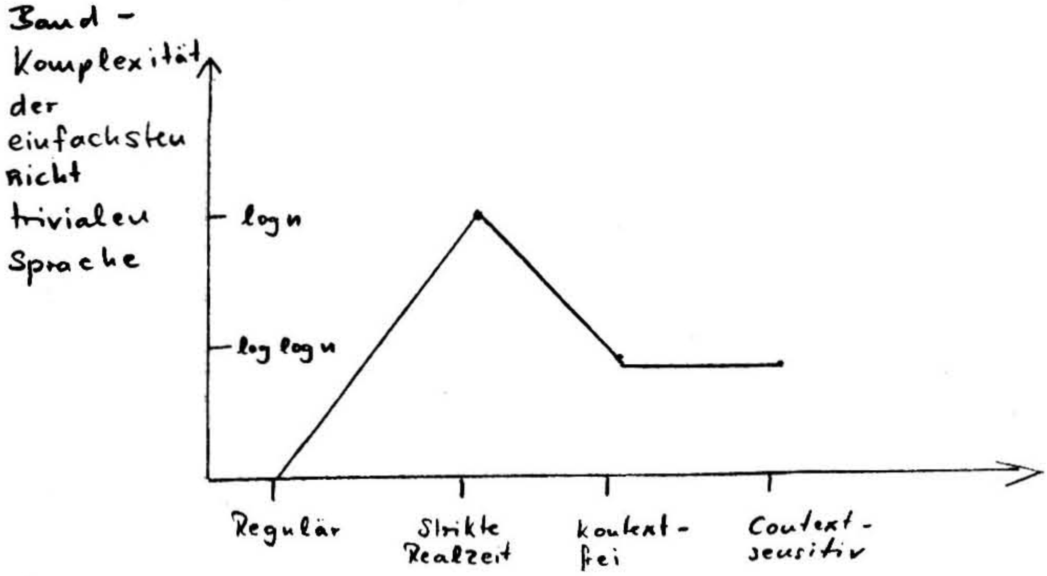
<u>Beweis:</u> Sei $L'' = \{a^pb^q; uv^pwx^qy \in L'\}$. Es ist leicht zu sehen, daß mit L' auch L" regulär wäre. L" ist aber wegen 1) und 2) nicht-regulär.

Wir haben also nun gezeigt, daß L stark nicht-regulär ist. Also benötigt das Wortproblem von L unendlich oft logarithmischen Speicher.

Mit einem ähnlichen Argument kann man zeigen, daß das Wortproblem für jede nicht-reguläre strikt deterministische Zählersprache logarithmische Platzkomplexität hat. Die strikt deterministischen Zählersprachen sind Sprachen, die durch einen deterministischen Zählerautomaten akzeptiert werden, der stehen bleibt, falls der Speicher einmal leer wird.

6. Schwach wachsende Band-konstruierbare Funktionen

Im vorhergehenden Abschnitt zeigten wir, daß das Wortproblem für nicht-reguläre strikte Realzeitsprachen mindestens logarithmische Bandkomplexität hat. Harrison und Havel [3] gaben eine einfache grammatikalische Charakterisierung dieser Sprachklasse an. Die erzeugenden Grammatiken sind eine Teilmenge der kontext-freien. Unser Resultat liefert also folgendes Paradox: Um eine ganz einfache Sprache (= eine Sprache der Bandkomplexität log log n) erzeugen zu können, benötigt man einen relativ starken Erzeugungsmechanismus.



Wir wollen im folgenden versuchen, diesen Antagonimus aufzuklären.

Prinzipiell sind bei einer platzbeschränkten TM zwei Arbeitsweisen denkbar.

(1) Die Größe des Arbeitsspeichers wird am Anfang der Rechnung in Abhängigkeit von der Länge der Eingabe bestimmt. Für Worte gleicher Länge ist der Speicherbedarf also gleich, die Bandomplexität ist durch eine band-konstruierbare Funktion gegeben.

(2) Der Speicherbereich wird in Abhängigkeit vom bisherigen Verlauf der Rechnung dynamisch erweitert. Das Eingabewort steuert also die

dynamische Erweiterung des Speicherbereichs. Wir wollen diese Strategie an einem Beispiel erläutern.

Beispiel:

$W = \{ \# \beta(1) \# \beta(2) \# \beta(3) \# \ldots \# \beta(n) \# \; ; \quad n \text{ natürliche Zahl } \}$;
dabei ist $\beta(i)$ die Binärdarstellung der natürlichen Zahl i.

W kann mit O(log log n) Speicherplatz akzeptiert werden: Dazu überprüft man sequentiell von links nach rechts, ob die Zeichenreihen zwischen den Markierungen aufeinander folgende Binärzahlen sind. Dazu braucht man die beiden Worte nur Bitweise vergleichen; in einem Zähler (in Binärdarstellung) speichert man die Position des Bits, das gerade verglichen wird. Jedesmal wenn die Länge des Zählers nicht mehr ausreicht, verlängert man ihn um ein Bit. Das Eingabewort steuert also die dynamische Verlängerung des Zählers.

Im ersten Fall ist die Bandkomplexität durch eine band-konstruierbare Funktion gegeben.

Def.: Eine Funktion $f : \mathbb{N} \to \mathbb{N}$ heißt band-konstruierbar, wenn es eine eine off-line deterministische TM gibt, die an der Eingabe o^n genau f(n) Speicherzellen benutzt.

Es gilt nun der folgende Satz:

Satz: Sei f band-konstruierbar und sei $\liminf_{n \to \infty} \frac{f}{\log \log n} > o$.

Dann gilt sogar $\liminf_{n \to \infty} \frac{f}{\log n} > o$.

Beweis:
Wir benutzen ein Lemma von Jan Messerschmidt [7] .

Lemma: Es gibt eine natürliche Zahl M, so daß für alle $n,m \in \mathbb{N}$ gilt

$$\left.\begin{array}{l} n + m > M \\ n = m \end{array}\right\} \iff n \equiv m \pmod k \quad \text{für alle } k \le 4 \log (n+m)$$

Beweis: Sei $n \equiv m \pmod k$ für alle $k \le 4 \log (n+m)$.

Dann ist $n \equiv m \pmod P$, wo P das Produkt aller Primzahlen $\le 4 \log (n+m)$ ist.

Nach dem Satz von Gauss gibt es ungefähr $s/\log s$ Primzahlen $\le s$.

Also gilt:

$$\begin{aligned}
\prod_{\substack{p \le s \\ p \text{ Primzahl}}} p &\ge (s/\log s)! \\
&\ge (s/2 \log s)^{(s/2 \log s)} \\
&\ge 2^{(s/2 \log s) \log (s/2 \log s)} \\
&\ge 2^{(s/2 \log s) \cdot (\log s - \log 2 \log s)} \\
&\ge 2^{s/4}
\end{aligned}$$

für hinreichend große s. Mit $s = 4 \log (n+m)$ erhalten wir $P \ge n+m$

Sei also nun f band-konstruierbar mit $\liminf f(n) / \log \log n = c > 0$.

Wir definieren eine Sprache $L \subseteq a^* b^*$ der Bandkomplexität f.

Eingabe $a^n b^m$

(1) Grenze den Arbeitsbereich der Länge $l = f(n+m)$ auf dem Band ab.

(2) Teste, ob für alle $k \leq \lceil 4^{1/c} \rceil^{l}$ gilt:
$n \equiv m \pmod{k}$.

(3) Falls ja, dann akzeptiere $a^n b^m$.

Sei M so gewählt, daß $f(n+m) \geq c \cdot \log\log(n+m)$ für $n+m \geq M$. Dann gilt für $n+m \geq \max(M, 2^4)$

$$a^n b^m \in L \Longrightarrow n \equiv m \pmod{k} \text{ für alle } k \leq \lceil 4^{1/c} \rceil^{f(n+m)}$$

$$\Longrightarrow n \equiv m \pmod{k} \text{ für alle } k \leq \lceil 4^{1/c} \rceil^{c \cdot \log\log(n+m)}$$

$$= (\log(n+m))^2$$

$$\Longrightarrow n \equiv m \pmod{k} \text{ für alle } k \leq 4 \cdot \log(n+m)$$

$$\Longrightarrow n = m$$

Demnach ist L im wesentlichen die Sprache $a^n b^n$. Wir wissen aber bereits, daß das Wortproblem für $a^n b^n$ die Bandkomplexität $\log n$ hat. Also gilt $\liminf f(n)/\log n > 0$.

An diesem Beweis sieht man folgendes. Um die Sprache $\{a^n b^n;\ n \in \mathbb{N}\}$ zu erkennen, braucht man bei der Eingabe $a^n b^m$ nur $n \equiv m \pmod{k}$ für alle $k \leq 4 \cdot \log(n+m)$ nachzuprüfen.

Dies wird man so durchführen, daß man sukzessive größere Werte von k durchprobiert und auch sukzessive den Speicher erweitert. Falls $n \neq m$ ist, wird man $n \not\equiv m \pmod{k}$ für ein $k \leq 4 \cdot \log(n+m)$ feststellen. In diesem Fall kommt man also mit einem Speicherplatz der Größe $O(\log\log(n+m))$ aus. Wenn allerdings $n = m$ ist, dann besteht das Paar (n,m) sämtliche Teste der Form $n \equiv m \pmod{k}$. Man muß also auch ein Abbruchkriterium vorsehen.

Die Bandkomplexität dieses Abbruckkriteriums ist $O(\log(n+m))$.

Da der Logarithmus eine band-konstruierbare Funktion ist, erreicht man die Bandkomplexität $O(\log(n+m))$ auch mit der ersten Strategie. In diesem Fall bringt also die Strategie der dynamischen Erweiterung des Speichers keine Verringerung der Bandkomplexität: Die Struktur des Eingabeworts $a^n b^m$ ist zu einfach, um die dynamische Erweiterung des Speichers effizient zu steuern.

7. Eine Sprache der Bandkomplexität O(log log n) über einem einelementigen Alphabet

In diesem Abschnitt untersuchen wir die Frage, ob die Aussagen in den vorhergehenden Abschnitten best möglich sind. Wir zeigen, daß man in unserem Hauptsatz die Voraussetzung

"L kontextfrei" nicht weglassen kann.

Sei $L = \{a^n;$ die kleinste Zahl q, die n <u>nicht</u> teilt, ist eine Zweierpotenz$\}$.

Für $n \in \mathbb{N}$ sei q(n) die kleinste Zahl, die n nicht teilt. Nach dem Lemma in Abschnitt 6 gilt: q(n) = O(log n). Der folgende Algorithmus akzeptiert L.

$q \leftarrow 1;$

<u>repeat</u> $q \leftarrow q+1;$ $m \leftarrow n$ mod q <u>until</u> $m \neq 0$;
<u>if</u> q ist eine Zweierpotenz <u>then</u> accept
<u>else</u> reject.

Für festes q kann die Repeat-Anweisung in Platz O(log q) ausgeführt werden. Da q(n) der maximale q Wert ist und q(n) = O(log n), genügt also O(log log n) Platz, um L zu erkennen.

Es bleibt zu zeigen, daß L nicht regulär ist. Die Menge der Zahlen mit $q(N) = 2^k$ ist durch

$$L_k = m_k \mathbb{N} - 2^k \mathbb{N}$$
$$= m_k + 2m_k \mathbb{N}$$

gegeben, wo m_k das kleinste gemeinsame Vielfache der Zahlen $1,\dots,2^k-1$ ist. Dann ist $L = \bigcup L_k$.

Nehmen wir nun an, daß L regulär ist. Dann ist L endliche Vereinigung von periodischen Mengen: $L = \bigcup_{i \in J} s_i + t_i \mathbb{N}$ für J endlich. Für hinreichend großes k wird m_k von jedem t_i geteilt. Sei also k genügend groß. Wegen $m_k \in L$ gilt auch $m_k \in s_i + t_i \mathbb{N}$ für ein $i \in J$.

Wegen $t_i | m_k$ gilt also auch $2m_k \in s_i + t_i \mathbb{N} \subseteq L$. Für unendlich viele k gibt es aber Primzahlen zwischen 2^k und 2^{k+1}; für solche k ist $2m_k \notin L$. Widerspruch.

<u>Satz:</u> $L = \{a^n$; die kleinste Zahl q, die n nicht teilt, ist eine Zweierpotenz$\}$ ist eine nicht-reguläre Menge der Platzkomplexität O(log log n).

Für unendlich viele n ist natürlich q(n) sehr klein; etwa q(n) = 2 für alle ungeraden n. Die Bandkomplexität der L erkennenden Prozedur erfüllt also nicht die Voraussetzungen des Satzes von Abschnitt 6.

Referenzen

[1] Ginsburg, S.: The Mathematical Theory of Context-Free Languages, McGraw-Hill 1972

[2] Lewis, P.M., Hartmanis, J. and Stearns, R.E.: Memory bounds for the recognition of context-free and context-sensitive languages, IEEE Conference Record on Switching Circuit Theory and Logical Design, 191-202, 1965

[3] Harrison, M., Havel, J.M.: On a Family of Deterministic Grammars. 1st Colloquium on Automata, Languages and Programming, Paris, 1972.

[4] Hopcroft, J.W. and Ullman, J.D.: Formal languages and their relation to automata, Addison-Wesley, Reading, Mass., 1969

[5] Hotz, G. and Messerschmidt, J.: Dyck-Sprachen sind in Bandkomplexität log n analysierbar, Techn. Bericht der Univ. des Saarlandes, 1974

[6] Mehlhorn, K.: Bracket-languages are recognizable in logarithmic space, Techn. Bericht der Univ. des Saarlandes, 1975

[7] Messerschmidt, J.: Persönliche Mitteilung

[8] Monien, B.: Transformation methods and their applications to complexity problems, 2-te GI-Fachtagung, Kaiserslautern, 1975.

[9] Sudborough, I.H.: On tape-bounded complexity classes and multihead finite automata, 14th IEEE SWAT Conference, 138-144, 1973

[10] Valiant, L.G.: Decision procedures for Families of Deterministic Pushdown Automata, Ph. D Thesis of Warwick, 1973