Computing 7, 264—274 (1971)

Tabulator-Turingmaschine und Komplexität

Von

R. Weicker, Berlin

(Eingegangen am 1. Februar 1971)

Zusammenfassung — Summary

Tabulator-Turingmaschine und Komplexität. Es wird ein „Tabulator-Turingmaschine" genanntes Modell einer Turingmaschine definiert, die in einem Schritt mehrere Felder ihres Bandes überspringen kann und dadurch einer realen Datenverarbeitungsanlage näher kommt als eine Turingmaschine der üblichen Definition. Es wird gezeigt, daß diese Maschine zu anderen Komplexitätsklassen führen kann als eine normale mehrbändige Turingmaschine. Als Anwendung wird die Komplexität verschiedener Erkennungsverfahren für kontextfreie Sprachen angegeben.

Tabulator-Turing Machine and Complexity. A new model of a Turing Machine, called "Tabulator Turing Machine", is defined. It can shift its head over several squares of the tape and is therefore more similar to a real computer than a normal Turing Machine. The complexity classes defined by this machine are different from those defined by a normal multitape Turing Machine. This device is applied to measure the complexity of several recognition procedures for context free languages.

1. Zum Begriff der Komplexität

Bei den Untersuchungen, die in den letzten Jahren in wachsender Zahl unter dem Stichwort „Komplexität" veröffentlicht wurden, geht es meist darum, ein Maß für die Effektivität eines Algorithmus zu finden, vor allem im Hinblick auf Begriffe wie Rechenzeit oder Speicherplatz, die bei Datenverarbeitungsanlagen eine Rolle spielen. Dabei wird in der Regel, ausgehend von einer Arbeit von Hartmanis und Stearns [4] aus dem Jahre 1965, eine Turingmaschine (abgekürzt: TM) zugrunde gelegt, da sie das allgemeinste Modell einer Datenverarbeitungsanlage darstellt und nach der Church-Turingschen These jedes überhaupt mathematisch formulierbare Verfahren sich durch eine TM realisieren läßt. Die Zahl der Schritte, die eine — meist mehrbändige — TM zur Ausführung eines Algorithmus benötigt, und die Zahl der Felder auf ihren Bändern, die sie im Verlauf der Operationen beschreibt, werden, wenn sie sich als Funktionen der Länge der Ein- oder Ausgabewörter darstellen lassen, als Zeit- bzw. Bandkomplexität des Algorithmus bezeichnet.

Auf diese Weise gelangt man zu Komplexitätsaussagen, die unabhängig von einer speziellen Datenverarbeitungsanlage sind. Sie haben jedoch den Nachteil, daß sie unter Umständen nicht mehr brauchbar sind, wenn es um die konkrete Realisierung eines Algorithmus in der

Praxis geht. Insbesondere benötigt eine TM oft viele Schritte nur zum Verschieben ihres Lesekopfes auf dem Band, um an gespeicherte Information heranzukommen, während bei einer Datenverarbeitungsanlage normalerweise eine Speicherzelle über ihre Adresse direkt angesprungen werden kann. Deshalb wird im folgenden ein Komplexitätsmaß für eine TM so definiert, daß reine Verschiebeoperationen keine Rolle spielen. Eine Adressierung mit festen Adressen für die gespeicherte Information, an die man zunächst denken könnte und die der Praxis noch näher käme, hätte zur Folge, daß das Konzept einer TM mit potentiell unendlichem Speicherband, aber einer endlichen Maschinentafel, nicht beibehalten werden könnte. Ein Ausweg bietet sich durch die Definition einer Turingmaschine, die außer dem normalen Bandalphabet noch eine endliche Anzahl von Marken besitzt, die jeweils auf ein Feld des Bandes gesetzt werden und auch wieder gelöscht werden können. Auf ein solches markiertes Feld soll der Lesekopf in *einem* Schritt der Maschine springen können, die Maschine soll deshalb „Tabulator-Turingmaschine" heißen. Damit ergibt sich eine gewisse Analogie zum Prinzip der assoziativen Speicherung, die Tabulator-Turingmaschine könnte als abstraktes Modell einer Datenverarbeitungsanlage mit assoziativem Speicher angesehen werden. Im folgenden wird eine solche Tabulator-Turingmaschine genauer definiert, und es werden einige charakteristischen Eigenschaften einer solchen Maschine untersucht. Dabei handelt es sich hier um eine On-Line-Maschine mit einem Eingabe- und einem Arbeitsband, wie es zur Sprachenerkennung am günstigsten ist.

2. Definition der Tabulator-Turingmaschine

Eine Tabulator-Turingmaschine (abgekürzt: TTM) ist ein 8-Tupel $T = (Z, A, X, M, Z_M, Z_F, z_0, \delta)$ mit

Z endliche Menge von Zuständen
A Eingabealphabet
X Bandalphabet, mit $\mathfrak{S}$ und $\mathfrak{C}$ als Begrenzungszeichen
M endliche Menge von Marken, $M = \{m_1, \ldots, m_r\}$
$Z_M \subseteq Z$ Zustände, die zum Sprung auf eine Marke führen, wobei $Z_M = \{z_1, \ldots, z_r\}$, also $|Z_M| = |M|$
$Z_F \subseteq Z$ Endzustände
$z_0 \in Z$ Anfangszustand
$\delta\colon Z \times A \times X \to Z \times X \times \{0, 1\} \times \{-1, 0, 1\} \times \{-1, 0, 1\}^r$ Überführungstransformation, die auf einer (echten) Teilmenge von $Z \times A \times X$ erklärt und dort eindeutig ist.

Die Arbeitsweise der Maschine ergibt sich, indem die Abbildung

$$\delta\,(z, a, x) = (z', x', d_1, d_2, j_1, \ldots, j_r)$$

folgendermaßen als Maschinenoperation gedeutet wird:

Die Maschine besitzt ein Eingabeband mit einem Lesekopf, der nur in einer Richtung bewegt werden kann, und ein Arbeitsband mit einem

Lese- und Schreibkopf. Das Arbeitsband ist in zwei Spuren unterteilt, eine für den eigentlichen Bandinhalt, die Symbole aus X (Symbolspur), und eine für die Marken aus M (Adressenspur). Dabei kann ein Feld mehrere verschiedene Marken gleichzeitig, maximal also r Marken enthalten. z sei der augenblickliche Zustand der Maschine, a das Symbol unter dem Lesekopf des Eingabebandes, x das Symbol unter dem Lesekopf des Arbeitsbandes. Durch die Abbildung $\delta(z, a, x) = (z', x', d_1, d_2, j_1, \ldots, j_r)$ wird der Schritt beschrieben, in dem die Maschine vom Zustand z in den Zustand z' übergeht, gleichzeitig wird der Lesekopf auf dem Eingabeband in Abhängigkeit von z, a und x bewegt, bei $d_1 = 1$ rückt er ein Feld weiter, bei $d_1 = 0$ bleibt er für diesen Schritt stehen. Auf dem Arbeitsband wird in der Symbolspur das Symbol x durch das Symbol x' ersetzt. In der Adressenspur enthalte das Feld unter dem Lesekopf einige der Marken $m_1, \ldots, m_r$. Für jede Marke m_i $(1 \leqslant i \leqslant r)$ gibt dann j_i an, ob m_i gelöscht wird (im Fall $j_i = -1$) oder gesetzt wird (im Fall $j_i = 1$). Im Fall $j_i = 0$ ändert sich in Bezug auf die Marke m_i nichts, sie bleibt im Feld unter dem Lesekopf genau dann stehen, wenn sie sich vorher dort befand. Die Fortbewegung des Lesekopfes auf dem Arbeitsband hängt davon ab, ob $z \in Z_M$ gilt: Im Fall $z \notin Z_M$ rückt der Lesekopf, wie üblich, ein Feld nach links, rechts, oder bleibt stehen, je nachdem ob $d_2 = -1$, 1 oder 0 ist. Andernfalls sei $z = z_i \in Z_M$. Der Lesekopf rückt dann in diesem einen Schritt der TTM so weit nach links (im Fall $d_2 = -1$) bzw. nach rechts (im Fall $d_2 = 1$), bis sich auf der Adressenspur des Bandes die dem Zustand z_i entsprechende Marke m_i oder ein Begrenzungszeichen unter dem Lesekopf befindet. Eine Marke m_i kann auf dem Band mehrfach vorkommen, der Sprung des Lesekopfes erfolgt jeweils auf die nächste Marke m_i in der durch d_2 bestimmten Richtung. Wie üblich, soll δ der Einschränkung unterliegen, daß ein Begrenzungszeichen vom Lesekopf höchstens weitergeschoben, nicht aber überschritten wird.

Zur formalen Beschreibung des Verhaltens einer TTM kann in der üblichen Weise eine *Konfiguration* definiert werden als eine Folge der Form

$$\alpha = u!w\,\mathrm{S}\,M^{(1)}\,x^{(1)}\ldots z\,M^{(i)}\,x^{(i)}\ldots M^{(n)}\,x^{(n)}\,\mathrm{C}$$

mit $u, w \in A^*$, $M^{(j)} \subseteq M$, $x^{(j)} \in X \setminus \{\mathrm{S}, \mathrm{C}\}$ und $z \in Z$. Dadurch wird beschrieben, daß die TTM sich im Zustand z befindet, auf dem Eingabeband steht die Folge $u\,w$, auf dem Arbeitsband jeweils die Markenmenge $M^{(j)}$ (die auch leer sein kann) und das Symbol $x^{(j)}$, die Leseköpfe befinden sich über dem ersten Symbol von w bzw. über dem Paar $(M^{(i)}, x^{(i)})$. Damit lassen sich die Bewegungen der TTM formal als Aufeinanderfolgen von Konfigurationen beschreiben, was durch $\alpha \vdash \beta$ wiedergegeben wird.

Hier seien nur die Operationen angegeben, die bei einer normalen TM nicht vorkommen:

1. $u!a\,w\,\mathrm{S}\,M^{(1)}\,x^{(1)}\ldots z\,M^{(i)}\,x^{(i)}\ldots M^{(n)}\,x^{(n)}\,\mathrm{C}$
$$\vdash u'!w'\,\mathrm{S}\,M^{(1)}\,x^{(1)}\ldots z'\,M^{(i-s)}\,x^{(i-s)}\ldots M^{(i-1)}\,x^{(i-1)}\,M'\,x'$$
$$M^{(i+1)}\,x^{(i+1)}\ldots M^{(n)}\,x^{(n)}\,\mathrm{C},$$

falls $\delta(z, a, x^{(i)}) = (z', x', ., -1, \ldots)$, $z = z_j \in Z_M$, $m_j \in M^{(i-s)}$ und $m_j \notin M^{(i-1)}, M^{(i-2)}, \ldots, M^{(i-s+1)}$;

bzw. im Fall $m_j \notin M^{(i-1)}, \ldots, M^{(1)}$:

$$\vdash u'!w'\, z'\, \mathbf{S}\, M^{(1)}\, x^{(1)} \ldots M^{(i-1)}\, x^{(i-1)}\, M'\, x'\, M^{(i+1)}\, x^{(i+1)} \ldots M^{(n)}\, x^{(n)}\, \mathbf{C}$$

(Sprung nach links).

2. $u!a\,w\,\mathbf{S}\, M^{(1)}\, x^{(1)} \ldots z\, M^{(i)}\, x^{(i)} \ldots M^{(n)}\, x^{(n)}\, \mathbf{C}$

$$\vdash u'!w'\,\mathbf{S}\, M^{(1)}\, x^{(1)} \ldots M^{(i-s)}\, x^{(i-s)}\, M'\, x'\, M^{(i+1)}\, x^{(i+1)} \ldots$$
$$\ldots z'\, M^{(i+s)}\, x^{(i+s)} \ldots M^{(n)}\, x^{(n)}\, \mathbf{C},$$

falls $\delta(z, a, x^{(i)}) = (z', x', ., 1, \ldots)$, $z = z_j \in Z_M$, $m_j \in M^{(i+s)}$ und $m_j \notin M^{(i+1)}, \ldots, M^{(i+s-1)}$;

bzw. im Fall $m_j \notin M^{(i+1)}, \ldots, M^{(n)}$:

$$\vdash u'!w'\,\mathbf{S}\, M^{(1)}\, x^{(1)} \ldots M^{(i-1)}\, x^{(i-1)}\, M'\, x'\, M^{(i+1)}\, x^{(i+1)} \ldots M^{(n)}\, x^{(n)}\, z'\, \mathbf{C}$$

(Sprung nach rechts).

3. Die an die Stelle von $M^{(i)}$ tretende neue Markenmenge M' ist

$$M' = (M^{(i)} \cup \{m_k \mid j_k = 1\}) \setminus \{m_k \mid j_k = -1\}, \text{ wenn}$$
$$\delta(z, a, x^{(i)}) = (., ., ., ., ., j_1, \ldots, j_r).$$

Wie üblich werden die transitive Hülle der Relation „$\vdash$", Endkonfiguration und Bandmenge einer TTM definiert: Für zwei Konfigurationen α und β gilt $\alpha \vdash^* \beta$, wenn es Konfigurationen $\alpha_1, \ldots, \alpha_m$ mit $\alpha = \alpha_1$, $\beta = \alpha_m$ und $\alpha_i \vdash \alpha_{i+1}$ $(i = 1, \ldots, m-1)$ gibt. *Endkonfiguration* einer TTM ist eine Konfiguration $\alpha = u!a\,w\,\mathbf{S} \ldots z\, M'\, x \ldots \mathbf{C}$, wenn $\delta(z, a, x)$ nicht definiert ist. Gilt für eine Kette $w \in A^*$ und eine TTM T $!w\,\mathbf{S}\, z_0\, \mathbf{C} \vdash^* \alpha$, wobei $\alpha = w!\,\mathbf{S} \ldots z \ldots \mathbf{C}$ Endkonfiguration, so heißt w von T angenommen, falls $z \in Z_F$, und abgelehnt, falls $z \notin Z_F$; die Menge der von T angenommenen Ketten aus A^* wird als die *Bandmenge* von T bezeichnet. Maschinen, die die gleiche Bandmenge besitzen, heißen *äquivalent.*

Auch die Komplexität (hier wird nur die Zeitkomplexität betrachtet, da für die Bandkomplexität eine TTM nichts Neues bringt) wird analog zum Vorgehen bei einer TM definiert: Eine TTM hat die *Komplexität* $T(n)$, wenn sie jede Kette von n Symbolen nach höchstens $T(n)$ Schritten annimmt oder ablehnt.

Man kann selbstverständlich die TTM auch in leichter Abänderung der Definition als Off-Line-Maschine ohne eigenes Eingabeband entwerfen, ebenso ist eine Erweiterung zu einer TTM mit mehreren Arbeitsbändern analog dem Vorgehen bei einer normalen TM möglich. Es wird sich jedoch zeigen, daß dies gegenüber einer einbändigen TTM keine wesentlichen Vorteile bietet.

Die hier definierte Tabulator-Turingmaschine entspricht in ihrem Verhalten der Turingmaschine, die C.-P. Schnorr, mit etwas anderer Terminologie, in [8] definiert hat. Eine ähnliche Maschine, allerdings mit zwei wesentlichen Einschränkungen, wurde von M. J. Fischer und A. L. Rosenberg in [2] untersucht, auf sie wird im nächsten Abschnitt noch näher eingegangen. Der Name „Tabulator-Turingmaschine" wurde der Maschine in Anlehnung an die Einführung eines „Tabulator Stack Automaton" von S. R. Kosaraju in [6] gegeben, er beschreibt wohl am anschaulichsten ihre wesentliche Eigenschaft.

3. Eigenschaften der Tabulator-Turingmaschine

Bei jeder neuen Konstruktion eines Automaten ist die Frage naheliegend, wie sich seine Bandmenge zu der Bandmenge bekannter Automatentypen verhält. Für die TTM gilt

Satz 1. *Zu jeder TTM gibt es eine äquivalente TM und umgekehrt.*

Beweis. Daß jede TM sich als Spezialfall einer TTM mit $M = Z_M = \varnothing$ auffassen läßt, ist nach der Konstruktion einer TTM sofort klar. Umgekehrt aber läßt sich zu jeder TTM $T = (Z, A, X, M, Z_M, Z_F, z_0, \delta)$ eine TM $T' = (Z', A, X', Z_F, z_0, \delta')$ konstruieren, die als Bandalphabet $X' := \mathfrak{P}(M) \times X$ hat. Sie geht immer dann, wenn der Lesekopf von T einen Sprung auf eine Marke m macht, zunächst in einen Zwischenzustand über, ihr Lesekopf läuft in einzelnen Schritten in der entsprechenden Richtung, bis er auf dem Band ein Element aus $\mathfrak{P}(M) \times X$, also eine Marken-Symbol-Kombination findet, in der m enthalten ist, oder bis er auf ein Begrenzungszeichen stößt; dann findet der eigentliche Zustandsübergang statt. Schritte ohne Sprung auf eine Marke werden unverändert übernommen. Somit simuliert T' jeden einzelnen Schritt von T, ist also äquivalent zu T.

Während *Satz 1* zeigt, daß — wie zu erwarten war — die Menge der erkannten Sprachen sich durch eine TTM nicht ändert, weist das folgende Beispiel darauf hin, daß die dann nötige Anzahl von Schritten durch eine TTM unter Umständen erheblich verringert werden kann.

Gegeben sei die Sprache $\{w\, w^T \mid w \in A^*\}$ (für $w = a_1 \ldots a_k$ bezeichne w^T die Kette $a_k \ldots a_1$), sie soll durch eine einbändige Turingmaschine erkannt werden.

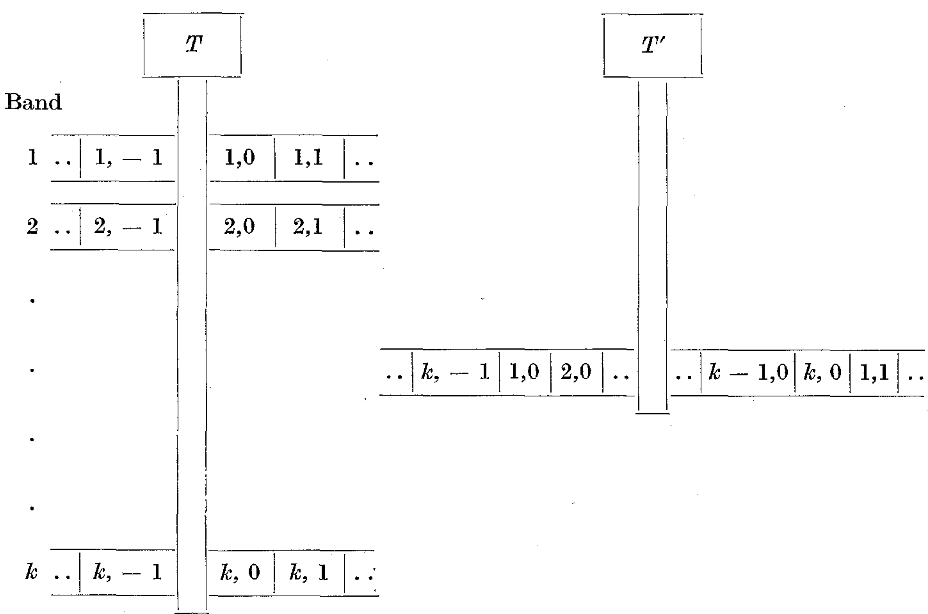
Dazu liegt es nahe, eine Maschine zu konstruieren, die eine vorgelegte Kette auf ihr Arbeitsband überträgt und dann von den Enden her untersucht. Sie beginnt am einen Ende der Bandinschrift, liest ein Symbol, löscht es und geht in einen für dieses Symbol charakteristischen Zustand über. Darauf rückt der Lesekopf zum anderen Ende der Bandinschrift; wenn das dort vorgefundene Symbol das gleiche ist, wird es ebenfalls gelöscht, und mit dem nächstinneren Symbolpaar wird in der selben Weise verfahren. Läßt sich das Verfahren fortsetzen, bis das ganze Band gelöscht ist, so wird die gegebene Kette angenommen, sonst abgelehnt.

Ist n die Länge der vorgegebenen Kette, so benötigt eine normale TM zum Erkennen des äußersten Symbolpaares $n + 2$ Schritte, für das nächstinnere Paar n Schritte usw., insgesamt $n + 2 + n + \ldots + 4 = C\,n^2$ Schritte. Eine TTM dagegen setzt sich beim ersten Überlaufen Marken an die beiden Enden und benötigt dann zum Aufsuchen des jeweils nächsten zu prüfenden Symbols nur 2 Schritte, insgesamt also nur $C\,n$ Schritte.

Bei einer normalen TM hätte man freilich durch Übergang zu einer mehrbändigen Maschine die Schrittzahl auch herabsetzen können; wenn ein gegebener Algorithmus ausgeführt werden soll, hängt die Komplexität der Maschine wesentlich davon ab, ob es sich um eine einbändige oder eine mehrbändige TM handelt. Eine entsprechende Aussage gilt für die TTM nicht, wie der folgende Satz (vgl. [8], Lemma 2.5) zeigt:

Satz 2. *Zu jeder k-bändigen TTM T mit der Zeitkomplexität $T(n)$ gibt es eine äquivalente einbändige TTM T' mit der Zeitkomplexität $k(k + 3)\,T(n)$.*

Beweis. Zu T wird eine einbändige TTM T' konstruiert, deren Band in k Klassen unterteilt ist. In den Feldern der Klasse j $(1 \leqslant j \leqslant k)$, die jeweils k Schritte voneinander entfernt sind, wird die Bewegung des j-ten Bandes von T simuliert. Die Zuordnung wird durch die folgende Abbildung deutlich:



Die Maschine T' hat eine gegenüber T größere Zahl von Marken: Jeder Marke m_j von T entsprechen k Marken $m_{j1}, \ldots, m_{jk}$ von T', wobei die Marke m_{ji} genau dann gesetzt oder gelöscht wird, wenn dies mit m_j auf dem i-ten Band von T geschieht. Außerdem habe T' k Marken $l_1, \ldots, l_k$, die Marke l_i wird jeweils auf das Feld gesetzt, das dem unter dem unter dem Lesekopf befindlichen Feld des i-ten Bandes von T entspricht. Eine weitere Marke l schließlich befindet sich immer links neben dem äußersten linken beschriebenen Feld des Bandes von T', sie wird bei Bewegungen des Lesekopfes über das bisher letzte Feld hinaus jeweils weitergerückt.

Eine Operation auf dem i-ten Band von T kann dann von T' in der folgenden Weise simuliert werden: Der Lesekopf springt zunächst nach links auf das Feld mit der Marke l, dann nach rechts auf das Feld mit der Marke l_i, der Inhalt dieses Feldes wird entsprechend der Operation von T geändert. Der „Umweg" über die Marke l ist dabei notwendig, damit die Bewegungsrichtung des Lesekopfes eindeutig wird. Findet nun bei T eine Bewegung des i-ten Bandes statt, so wird die Marke l_i gelöscht und k Felder weiter links oder rechts, entsprechend der Bewegung bei T, wieder gesetzt (zusammen $k + 3$ Schritte); ist diese Bewegung ein Sprung auf die Marke m_j, so springt bei T' der Lesekopf auf das Feld mit der Marke m_{ji} und setzt dort die Marke l_i (zusammen 4 Schritte). Insgesamt ergibt sich damit, daß ein Schritt von T durch höchstens $k\,(k + 3)$ Schritte von T' simuliert wird, wobei das Arbeitsband von T' bei jedem Schritt genau den Inhalt der Arbeitsbänder von T enthält, T und T' also äquivalent sind.

Da jede mehrbändige TM auch als TTM aufgefaßt werden kann, folgt aus dem Satz insbesondere, daß eine mehrbändige TM sich durch eine einbändige TTM mit einer bis auf einen konstanten Faktor gleichen Komplexität ersetzen läßt. Die Frage liegt nahe, ob auch das Umgekehrte gilt, ob es also zu jeder einbändigen TTM der Komplexität $T\,(n)$ eine äquivalente mehrbändige TM der Komplexität $C\,T\,(n)$ mit einer von n unabhängigen Konstanten C gibt. Für eine etwas andere Art von Tabulator-Turingmaschine, wie sie von FISCHER und ROSENBERG definiert wurde, trifft diese Vermutung zu:

Satz 3. *Sei T eine (ein oder mehrbändige) TTM der Zeitkomplexität $T\,(n)$ mit den folgenden Einschränkungen:*

1. *Jede Marke kann auf einem Band höchstens einmal gesetzt werden.*
2. *Eine einmal gesetzte Marke kann nicht wieder entfernt werden.*

Dann gibt es zu T eine äquivalente mehrbändige TM normaler Art der Zeitkomplexität $C\,T\,(n)$ mit einer von n unabhängigen Konstanten C.

Der Beweis dieses Satzes, der recht aufwendig ist, findet sich in [2]. Im Gegensatz dazu läßt sich jedoch für die hier definierte Art einer TTM zeigen:

Satz 4. *Es gibt eine Menge M, für die gilt:*

1. *Ist M Bandmenge einer (ein- oder mehrbändigen) TM der Zeitkomplexität $T\,(n)$, so gilt*

$$T\,(n) \geqslant C_1 \frac{n^2}{(\log n)^2}\,.$$

2. *M ist Bandmenge einer einbändigen TTM der Zeitkomplexität $C_2\,n$ mit einer von n unabhängigen Konstanten C_2.*

Beweis. Die Menge M wird nach T. Kasami [5] und C. F. Hennie [3] definiert als die Menge der endlichen Folgen über dem Alphabet $\{0, 1, s, c\}$, die den folgenden Bedingungen genügt:

a) Das erste und das letzte Symbol der Folge ist s, die Folge besteht im übrigen aus einer Anzahl von Blöcken gleicher Länge, die ihrerseits Folgen über dem Alphabet $\{0, 1\}$ sind und durch s oder scs voneinander getrennt sind.

b) Die Anzahl der $0 - 1$-Blöcke ist größer als 2^k, wobei k die Länge eines einzelnen Blocks ist. Der 2^k-te und der $(2^k + 1)$-te Block sind durch scs voneinander getrennt, alle folgenden Blöcke durch einfaches s.

c) Ist w der letzte $0 - 1$-Block, so ist einer der ersten 2^k Blöcke der Block w^T.

Daß für diese Menge die Behauptung 1. gilt, ist die Aussage von *Lemma 1* aus [5], das auf einer in [3] ausführlich beschriebenen Beweisidee beruht. Andererseits kann, etwa in der folgenden Weise, eine einbändige TTM der Komplexität $C_2\, n$ entscheiden, ob eine vorgegebene Kette Element von M ist:

Sei n die Länge der Kette, sie wird zunächst auf das Arbeitsband der TTM übernommen (n Schritte). Nun kann die Kette noch einmal überlaufen, bei jedem neuen Symbol eines Blockes eine Marke auf einem weiteren (Kontroll-)Abschnitt des Bandes verschoben und dadurch festgestellt werden, ob die Bedingung a) erfüllt ist ($4\, n$ Schritte). Durch schrittweises Verdoppeln eines Abschnittes von zunächst 2 Symbolen kann weiter die Zahl 2^k (k = Länge eines Blocks) dargestellt werden (höchstens $2^{k+2} + 3\, k \leqslant 4\, n$ Schritte), sie wird dann mit der Zahl der Blöcke verglichen ($n + 2\, k \leqslant 2\, n$ Schritte) und damit Bedingung b) nachgeprüft. Schließlich kann bei einem weiteren Überlaufen des Bandinhalts, bei dem der Lesekopf nach jedem Schritt zu einem Vergleich an das Ende der Kette springt, Bedingung c) nachgeprüft werden (4 n Schritte). Insgesamt sind also zur Annahme oder Ablehnung einer Kette der Länge n höchstens $15\, n$ Schritte der TTM nötig, Behauptung 2. ist damit gezeigt.

Satz 4 zeigt, daß durch die Definition der TTM ein neues Komplexitätsmaß gefunden wurde, das nicht in allen Fällen mit dem üblichen, durch eine mehrbändige TM gegebenen Komplexitätsmaß übereinstimmt; daher erscheint es berechtigt, die TTM als eigenen Automatentyp neben der TM zu betrachten. Die TTM im Beweis zu *Satz 4* kann, wie man leicht einsieht, so definiert werden, daß zu jedem festen Zeitpunkt jede Marke nur einmal auf dem Band auftritt, daß sie also die in [4], S. 296 von Hartmanis und Stearns angegebene Bedingung erfüllt. Die dort aufgestellte Behauptung, daß eine solche Maschine in der gleichen Komplexitätsklasse liegt wie eine mehrbändige TM, die die gleiche Berechnung ausführt, kann daher in dieser Allgemeinheit nicht aufrecht erhalten werden, sie trifft zwar für eine Maschine des Fischer-Rosenbergschen Typs zu (*Satz 3*), nicht aber für die hier definierte TTM.

4. TTM-Komplexität von Erkennungsverfahren für kontextfreie Sprachen

Für verschiedene Erkennungsverfahren für kontextfreie Sprachen, sowohl vom Bottom-Up- als auch vom Top-Down-Typ, wurde in [9] die Komplexität bei Realisierung des Verfahrens durch eine TTM berechnet. Hier werden nur die Ergebnisse wiedergegeben, für eine ausführlichere Diskussion der Verfahren und die Einzelheiten der Berechnung sei auf [9] verwiesen.

Es wurden untersucht:

YOUNGER I	Bottom-Up-Verfahren nach YOUNGER für Grammatiken in CHOMSKYscher Normalform [11].
YOUNGER II	Verallgemeinertes Bottom-Up-Verfahren nach YOUNGER für allgemeine Grammatiken [10].
KASAMI / TORII I	Bottom-Up-Verfahren nach KASAMI/TORII für Grammatiken in Normalform [7].
KASAMI / TORII II	Verallgemeinertes Bottom-Up-Verfahren nach KASAMI/TORII für allgemeine Grammatiken.
EARLEY	Top-Down-Verfahren nach EARLEY für allgemeine Grammatiken [1].

Außer der TTM-Komplexität wird hier auch, soweit sie bekannt ist oder sich leicht erschließen läßt, die Komplexität nach zwei anderen Maßstäben angegeben, und zwar für eine Realisierung durch eine mehrbändige TM im Sinne von [4] und durch einen Random-Access-Automaten mit wahlfreiem Zugriff zum Speicher (jedoch mit Speicherzellen fester Größe) im Sinne von [7].

Die Komplexität der verschiedenen Verfahren ist dann aus der folgenden Tabelle abzulesen:

Verfahren	Komplexität bei Realisierung durch		
	mehrbändige TM	einbändige TTM	Random-Access-Maschine
YOUNGER I	$\sim n^3$	$\frac{1}{3}(h+1)\,n^3$	$\sim n^3$
YOUNGER II		$\sim n^{m+1}$	$\sim n^{m+1}$
KASAMI/TORII I	$\sim n^4$ [1]	$\frac{4}{3}(h+f)\,n^3$	$\sim n^2 \log n$
KASAMI/TORII II	$\sim n^4$ [1]	$\frac{8}{3}\,g\,(m-1)\,n^3$	
EARLEY	$\sim n^4$ [1]	$\frac{4}{3}(g^2 m + g)\,n^3$	$\sim n^3 \log n$ [2] $\sim n^2 \log n$

[1] Diese Komplexität ergibt sich, indem die in [9] beschriebene Realisierung für die TTM auf eine TM normaler Definition übertragen wird.

[2] Die erste Zeile gilt für mehrdeutige, die zweite für eindeutige Grammatiken; der Faktor $\log n$ kommt gegenüber [1] hinzu, wenn Speicherzellen mit fester Größe angenommen werden.

Die TTM-Komplexität ist dabei auf eine TTM bezogen, deren Alphabet auf 4 Zeichen normiert ist, die auftretenden Konstanten sind

h Zahl der Zwischensymbole (Nonterminals) der Grammatik,

g Zahl der Produktionen,

m Maximalzahl von Symbolen auf der rechten Seite einer Produktion,

f Zahl der Produktionen der Form $A \to BC$ (für Grammatiken in Normalform).

Betrachtet man die Komplexität als Maß für die Güte eines Verfahrens und stellt entsprechende Vergleiche an, so ist dabei zu beachten, daß sie bei den verschiedenen Verfahren verschieden zu bewerten ist: Beim Verfahren nach Younger gibt sie, dem systematischen Vorgehen entsprechend, die tatsächlich benötigte Zahl von Schritten an, bei den anderen Verfahren dagegen nur eine obere Schranke, die in der Anwendung meist weit unterboten wird. Aber abgesehen von diesen grundsätzlichen Schwierigkeiten einer Beurteilung der Komplexität zeigt die Tabelle doch charakteristische Eigenarten der Verfahren: Die Komplexität der Verfahren nach Kasami/Torii und Earley verringert sich umso mehr, je mehr sich das Modell der zur Berechnung der Komplexität verwandten Maschine einer realen Rechenanlage nähert, während beim Verfahren nach Younger sich die Komplexität nicht wesentlich verändert. Hier wird noch einmal deutlich, daß der Versuch berechtigt ist, auch andere Möglichkeiten zur Definition der Komplexität als nach dem ursprünglichen Vorgehen von Hartmanis und Stearns zu untersuchen.

Literatur

[1] Earley, J.: An Efficient Context-Free Parsing Algorithm. Comm. ACM **13,** 94—102 (1970).

[2] Fischer, M. J., und A. L. Rosenberg: Limited Random Access Turing Machines. IEEE Conf. Rec. of 1968 Ninth Ann. Symp. on Switching and Automata Theory, Schenectady, New York, 356—367.

[3] Hennie, C. F.: On-Line Turing Machine Computations. IEEE Trans. Electr. Comp. **EC—15,** 35—44 (1966).

[4] Hartmanis, J., und R. E. Stearns: On the Computational Complexity of Algorithms. Trans. Am. Math. Soc. **117,** 285—306 (1965).

[5] Kasami, T.: A Note on Computing Time for Recognition of Languages Generated by Linear Grammars. Inf. Contr. **10,** 209—214 (1967).

[6] Kosaraju, S. R.: Recognition of Context-Free and Stack Languages. IEEE Conf. Rec. of 1969 Tenth Ann. Symp. on Switching and Automata Theory, Waterloo, Ontario, 129—132.

[7] Kasami, T., und K. Torii: A Syntax-Analysis Procedure for Unambiguous Context-Free Grammars. Journal ACM **16,** 423—431 (1969).

[8] Schorr, C.-P.: Komplexität von Algorithmen mit Anwendung auf die Analysis. Universität Saarbrücken, Inst. Angew. Math. (1969).

[9] Weicker, R.: Erkennung von kontextfreien Sprachen durch Tabulator-Turingmaschinen. Arbeitsber. Inst. Math. Masch. Datenverarb., Univ. Erlangen, Bd. 3, Nr. 5 (1970).

[10] WOODS, W. A., und S. KUNO: A Recognition Procedure for Arbitrary Context-Free Grammars and the Upperbound of Recognition Time. Comp. Lab. Harvard Univ., Rep. No. NSF-17, Cambridge, Mass., X1—X26 (1966).

[11] YOUNGER, D. H.: Recognition and Parsing of Context-Free Languages in Time n^3. Inf. Contr. **10**, 189—208 (1967).

Dipl.-Math. Reinhold Weicker
Technische Universität Berlin
Informationsverarbeitung II
Ernst-Reuter-Platz 8, D-1 Berlin 10
Deutschland