Untersuchungen über Rado's Sigma-Funktion und eingeschränkte Halteprobleme bei Turingmaschinen

Von der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
genehmigte Dissertation

von Diplom-Mathematiker
Bruno Weimann
aus Mülheim-Ruhr

1973



Referent: Prof. Dr. H. Schiek

Korreferent: Prof. Dr. G. Hasenjaeger

Tag der mündlichen Prüfung: 5. November 1973

Herrn Prof. Dr. H. Schiek
danke ich herzlich für die persönliche Anteilnahme, die er mir und meiner Arbeit stets entgegengebracht hat.

Herrn Prof. Dr. W. Brauer
danke ich für manche nützliche Hinweise und Anregungen.

Dem Max-Planck-Institut für Kohlenforschung in Mülheim-Ruhr und seinem Direktor, Herrn Prof. Dr. G. Wilke, danke ich für die großzügig zur Verfügung gestellte technische Hilfe.

Inhaltsverzeichnis

I	Einleitung	1
II	Äquivalenz von Turingmaschinen	5
III	Konstruktionsverfahren für eine Teilmenge eines Repräsentantensystems $\mathcal{R}$	11
IV	Hinreichende Kriterien für das Nichthalten von Turingmaschinen	20
V	Ergebnisse über haltende Turingmaschinen mit bis zu 5 Befehlen	31
VI	Anmerkungen zur praktischen Durchführung	48
VII	Beweis der Lösbarkeit des Halteproblems bei nur schreibenden Turingmaschinen	60
	Literaturverzeichnis	68

I Einleitung

Die 1936 von A. Turing [1] erstmals definierte und 1952 von Kleene [2] präzisierte Turingmaschine hat dazu gedient, den Begriff des Algorithmus' exakt zu fassen und Probleme der Nichtberechenbarkeit von Funktionen zu lösen. T. Rado [3], [4] definierte 1962 für binäre Turingmaschinen, die im Gegensatz zur Turingmaschine nach Kleene in jedem Schritt das Arbeitsband um ein Feld verschieben müssen, die Funktionen $S(n)$ und $\Sigma(n)$, die auch Gegenstand der Untersuchung dieser Arbeit bilden.

$S(n)$ gibt die für binäre Turingmaschinen mit n Befehlen (bei Rado "Cards" genannt) maximal mögliche Laufzeit an, angegeben als Anzahl von durchgeführten Schritten bis zum Halt; $\Sigma(n)$ gibt die maximal mögliche Zahl von Einsen an (das Alphabet für binäre Turingmaschinen besteht, außer dem Leerzeichen "0", aus dem einzigen Zeichen "1"), die eine binäre Turingmaschine mit n Befehlen nach dem Halt auf dem Band hinterlassen kann. Bei beiden Funktionen wird vorausgesetzt, daß vor dem Start der Turingmaschine das Band leer ist. Die Konstruktion von Σ und S beruht auf dem Prinzip, daß jede nicht-leere endliche Menge ein größtes Element besitzt. Beide Funktionen sind, obwohl wohldefiniert (Rado: "exceptionally well-defined"), nicht berechenbar. Rado kleidete die Aufgabe der Berechnung von $\Sigma(n)$ in ein Spiel, das er "Busy-Beaver Game" nannte: Für vorgegebenes n ist eine Turingmaschine mit n Befehlen anzugeben. Sieger ist derjenige, dessen Turingmaschine bei Anwendung auf das leere Band hält und die meisten Einsen auf dem Band hinterläßt. Aus der Nichtberechenbarkeit von Σ und S läßt sich leicht die Unlösbarkeit des Halteproblems bei Turingmaschinen ableiten, die besagt, daß es keinen Algorithmus gibt,

der für eine beliebige Turingmaschine entscheidet, ob sie - angewandt auf das leere Band - hält oder nicht.

Im Anschluß an die Arbeiten von T. Rado [3], [4] wurden mehrere Versuche unternommen, $S(n)$ und $\Sigma(n)$ für kleine n direkt zu bestimmen oder aber durch Angabe bestimmter Turingmaschinen gute Näherungswerte zu erzielen. "Per Hand" lassen sich mit vertretbarem Aufwand nur die Werte für $n=1$ und $n=2$ bestimmen. Der erste und meines Wissens einzige Versuch, wenigstens für $n=3$ das Problem mit Hilfe eines Computers systematisch anzugehen, wurde von S. Lin in seiner Dissertation bei T. Rado 1963 unternommen (vgl. [5]) und führte zur Bestimmung von $\Sigma(3)$ und $S(3)$. Die Anzahl der möglichen Turingmaschinen mit 3 Befehlen wurde durch Einführung von Äquivalenzrelationen von 16 777 216 auf 82 944 zu untersuchende Turingmaschinen verringert, die dann mit Hilfe von Computerpogrammen in haltende und nichthaltende Turingmaschinen getrennt wurden. Bei 4C Turingmaschinen mußte diese Entscheidung "per Hand" getroffen werden, da die Computerprogramme dabei versagten. Das Verfahren von S. Lin auf Turingmaschinen mit 4 Befehlen angewandt würde von 25 600 000 000 möglichen auf 100 663 296 zu untersuchende Turingmaschinen führen, wobei etwa bei 15 000 das Halten bzw. Nichthalten "per Hand" entschieden werden müßte. Diese Tatsache bringt T. Rado [4] durch folgende Formulierung zum Ausdruck: "As regards $\Sigma(4)$, $S(4)$ the situation seems to be entirely hopeless at present.

Da die direkte Berechnung von $\Sigma(n)$ und $S(n)$ für $n \geq 4$ unmöglich schien, schloß sich der Versuch an, wenigstens eine untere Schranke für $\Sigma(n)$ und $S(n)$ anzugeben. Dies geschah in einer Arbeit von Milton W. Green [6]. Für $n \geq 6$ sind die von ihm angegebenen Werte immer noch die besten Näherungswerte. Die besten Werte für $n=4$ und $n=5$ kamen durch direkte Angabe von Turingmaschinen zustande.

In [7] ist zusammengefaßt, was für die ersten n bekannt ist:

	1	2	3	4	5	6	7	8
$\Sigma(n)$	=1	=4	=6	$\geq$13	$\geq$17	$\geq$35	$\geq$22 961	$\geq 3\cdot(7,3^{92}-1)/2$
$S(n)$	=1	=6	=21	$\geq$107				

Nach der von Milton W. Green gegebenen Abschätzung verlor die mit den Funktionen Σ und S zusammenhängende Problematik etwas an allgemeinem Interesse. Die vorliegende Arbeit wurde durch eine Vorlesung von W. Brauer (vgl. auch [8]) angeregt. In ihr ging es zunächst darum, mit Hilfe der Ansätze bei S. Lin und T. Rado [5] und mit Computerunterstützung die Funktionswerte $\Sigma(4)$ und $S(4)$ zu bestimmen und für n=5 eine bessere Abschätzung als in [7] angegeben zu ermitteln. Als Ergebnis dieser Bemühungen kann obige Tabelle geändert werden in:

	4	5
$\Sigma(4)$	=13	$\geq$40
$S(4)$	=107	$\geq$992

Nach längerer Zeit der Beschäftigung mit binären Turingmaschinen mit wenigen Befehlen und nach vielen fehlgeschlagenen Versuchen, Ordnungsstrukturen in der Menge aller Turingmaschinen mit n Befehlen zu finden, wurde außerdem in dieser Arbeit der Versuch unternommen, zumindest einen Einblick in die überraschend vielfältigen Möglichkeiten zu gewähren, die eine relativ einfache Struktur wie etwa eine binäre Turingmaschine mit 4 Befehlen gestattet. Zu diesem Zwecke war es notwendig, recht umfangreiche Tabellen und Statistiken aufzuführen.

Darüber hinaus führt eine Untersuchung, die sich mit "unlösbaren Problemen" beschäftigt, nahezu zwanglos zu der Frage, ob nicht durch eine Einschränkung des Problems eine Lösung erzielt werden

kann. Es liegt nahe, statt der Menge aller Turingmaschinen nur Teilmengen von Turingmaschinen zu betrachten und zu versuchen, dort das Halteproblem zu lösen, während der Nachweis der Unlösbarkeit des Halteproblems für die Teilmenge der nicht-löschenden (einmal vorhandene Einsen auf dem Band dürfen nicht mehr gelöscht werden) binären Turingmaschinen indirekt in einem Beweis bei John E. Hopcroft und Jeffrey D. Ullman [9] enthalten ist, wird als Ergebnis eigener Bemühungen der Beweis geführt, daß das Halteproblem lösbar ist, wenn man sich auf nur schreibende (jeder Befehl schreibt eine Eins) binäre Turingmaschinen beschränkt.

Zum Schluß der Einleitung sei noch auf eine sprachliche Regelung hingewiesen. Es wird in der vorliegenden Arbeit nicht zwischen Turingprogramm und Turingmaschine unterschieden. Wird von einer Turingmaschine gesprochen, so ist damit eine Turingmaschine gemeint, die mit einem bestimmten Turingprogramm "geladen" ist. Wird von Befehlen gesprochen, so sind die Befehle dieses Turingprogramms gemeint. Außerdem wird statt von binären Turingmaschinen nur von Turingmaschinen gesprochen.

II Äquivalenz von Turingmaschinen

Def. 1: Für die vorliegende Arbeit wird die Turingmaschine in Anlehnung an Brauer/Indermark [8] wie folgt definiert:

Eine Turingmaschine besteht im wesentlichen aus einer Zentraleinheit, einem Band, das nach beiden Seiten beliebig verlängerbar ist, einem Lese- und Schreibkopf, sowie einem Befehlsspeicher, in dem in natürlicher Weise durchnumeriert eine Anzahl von Befehlen stehen, nach denen die Turingmaschine arbeitet. Das Band, das gleichzeitig als Ein- und Ausgabeeinheit dient, ist in Felder eingeteilt und zwar vom Feld 0 ausgehend nach links die Felder -1,-2,-3.... und nach rechts die Felder 1,2,3.... Der Lese- und Schreibkopf steht zu jeder Zeit über genau einem Feld des Bandes, dem sogenannten Arbeitsfeld. Die Zentraleinheit kann von den Befehlen im Befehlsspeicher gesteuert das Zeichen auf dem Arbeitsfeld lesen und in Abhängigkeit davon und von dem augenblicklich durchgeführten Befehl ein Zeichen auf das Arbeitsfeld schreiben. Ein Zeichen ("1") und das Leerzeichen ("0") stehen zur Verfügung. Grundsätzlich wird beim Schreibvorgang das Arbeitsfeld überschrieben. Mit 0 überschreiben heißt das Arbeitsfeld löschen. Weiter kann die Zentraleinheit wieder in Abhängigkeit vom gelesenen Zeichen und vom augenblicklich durchgeführten Befehl den Lese- und Schreibkopf um ein Feld nach rechts oder links verschieben. Ein Befehl ist von der Form (insgesamt sollen n Befehle vorhanden sein):

$$i\ 0\ z_1^i\ v_1^i\ n_1^i \qquad i \in \{1,\ldots,n\},\ n_j^i \in \{0,1,\ldots,n\}$$

$$i\ 1\ z_2^i\ v_2^i\ n_2^i \qquad z_j^i \in \{0,1\},\ v_j^i \in \{R,L\}$$

Dabei sei i die Nummer des Befehls; 0 und 1 (2. Stelle der beiden Zeilen) regeln in Abhängigkeit vom Inhalt des Arbeitsfeldes, welche Zeile (= Teilbefehl) durchgeführt wird, d.h., steht auf dem Arbeitsfeld eine 0, so wird beim Aufruf des i-ten Befehls der 1. Teilbefehl dieses Befehls ausgeführt, steht auf dem Arbeitsfeld eine 1, wird der 2. Teilbefehl ausgeführt. Z_j^i ist das Zeichen, das bei der Durchführung des j-ten Teilbefehls des i-ten Befehls auf das augenblickliche Arbeitsfeld geschrieben wird. V_j^i sind die durchzuführenden Verschiebeoperationen. R bedeutet: Verschieben des Lese- und Schreibkopfes um ein Feld nach rechts; entsprechend steht L für links. n_j^i ist die Nummer desjenigen Befehls (nicht Teilbefehl), der als nächster von der Zentraleinheit durchzuführen ist. Ist $n_j=0$, so hält die Zentraleinheit nach der Durchführung dieses Teilbefehls die Turingmaschine an. (Ein solcher Befehl heißt Haltebefehl.) Grundsätzlich wird die Turingmaschine mit dem 1. Befehl im Befehlsspeicher gestartet.

Lemma 1: Es gibt $(4(n+1))^{2n}$ verschiedene Turingmaschinen mit n Befehlen [3] .

Beweis: Pro Teilbefehl gibt es 4 Möglichkeiten für die Kombination von $Z_j^iV_j^i$ und n+1 Möglichkeiten für n_j^i. Da eine Turingmaschine mit n Befehlen 2n Teilbefehle hat, gibt es $(4(n+1))^{2n}$ verschiedene Möglichkeiten.

Bemerkung: Um eine Vorstellung über das Anwachsen dieser Zahl zu bekommen, seien hier die ersten 6 Zahlen angegeben:

n	Anzahl der möglichen Turingmaschinen
1	64
2	20.736

3	16.777.216
4	25.600.000.000
5	63.403.380.965.376
6	232.218.265.089.212.416

Es wird nun eine Äquivalenzrelation auf der Menge aller Turingmaschinen definiert, indem Operationen angegeben werden, die Turingprogramme ändern, ohne daß sich der Ablauf wesentlich ändert. Bezüglich des Ablaufs wird immer vorausgesetzt, daß vor dem Start das Band leer ist. Indem man dann im weiteren nur noch einen Repräsentanten aus jeder Äquivalenzklasse zu untersuchen hat, verringert sich die Zahl der zu untersuchenden Turingmaschinen mit bis zu 4 Befehlen auf 1.198.690.

Def. 2: Zwei Turingmaschinen mit n Befehlen sind äquivalent, wenn sie identisch sind oder sich mit Hilfe einer oder mehrerer der folgenden Operationen ineinander überführen lassen:

1. Operation: In allen n Befehlen wird ein R durch ein L und ein L durch ein R ersetzt.

2. Operation: Ersetzen von R durch L und L durch R in Teilbefehlen, die Haltebefehle sind.

3. Operation: Für eine Permutation σ auf $\{1,2,\ldots,n\}$, die die 1 fest läßt, wird folgendes durchgeführt: σ wird angewandt auf die Befehlsnummern und auf die Nummern der aufzurufenden Befehle mit Ausnahme der 0, d.h. auf die von 0 verschiedenen Zahlen, die an erster und letzter Stelle in jedem Teilbefehl stehen. Anschließend werden die Befehle so geordnet, daß die Befehlsnummern wieder in natürlicher Reihenfolge stehen, Beispiel: n=4, σ =(1)(2,4,3).

101R2	101R4	101R4
111L2	111L4	111L4
201L1	401L1	201R0
210L3 $\Longrightarrow$	410L2 $\Longrightarrow$	211L3
301R0	201R0	301R3
311L4	211L3	310R1
401R4	301R3	401L1
410R1	310R1	410L2

4. Operation: Numeriert man statt der Befehle die Teilbefehle durch, so erhält man bei n Befehlen Teilbefehle mit den Nummern 1 bis 2n. Der Ablauf einer Turingmaschine kann dann als Folge von Teilbefehlsnummern dargestellt werden. Die Menge aller Teilbefehlnummern, die in dieser Folge auftauchen, sei M. M kann echte Untermenge von $\{1,\ldots,2n\}$ sein. Operation 4 besteht darin, daß Teilbefehle, die eine Nummer aus der Differenzmenge $\{1,\ldots,2n\}\setminus M$ haben, in irgendeiner beliebigen, nur den allgemeinen Regeln für den Aufbau eines Befehls entsprechenden Weise geändert werden. Anders ausgedrückt: Operation 4 ändert Teilbefehle, die beim Ablauf der Turingmaschine nie aufgerufen werden.

Bemerkung: Die Menge M in Operation 4 kann nur effektiv bestimmt werden, wenn von der betreffenden Turingmaschine das Halten bzw. das Nichthalten nachgewiesen ist. Hält die Turingmaschine, so kann M durch Aufführung aller bis zum Halt benutzten Teilbefehle bestimmt werden; hält sie nicht, so kann dieses Nichthalten nur nachgewiesen werden, indem man nach einer bestimmten Schrittzahl Regeln angibt, nach denen sich der weitere Ablauf vollzieht. Dann

läßt sich aber auch angeben, welche Teilbefehle im weiteren Verlauf noch benutzt werden.

Lemma 2: Durch Definition 2 wird eine Äquivalenzrelation auf der Menge aller Turingmaschinen definiert.

Reflexivität und Transitivität sind klar. Da zu jeder Operation auch die Umkehroperation zulässig ist, gilt auch die Symmetrie.

Bemerkung: Die 4 Operationen ändern eine Turingmaschine insofern nur unwesentlich, als der Ablauf der geänderten Turingmaschine von der Stellung des Lese- und Schreibkopfes sowie von der Bandbelegung sich nicht unterscheidet oder spiegelbildlich zum Feld 0 ist. Ob bei der Durchführung des Haltebefehls der Lese- und Schreibkopf nach links oder nach rechts verschoben wird, sollte nicht als Änderung des Ablaufs angesehen werden. Man könnte eine Äquivalenzrelation auch folgendermaßen definieren: Zwei Turingmaschinen sind äquivalent, wenn sich ihre Abläufe nicht oder nur unwesentlich, in der oben angegebenen Art, unterscheiden. Dabei zeigt es sich aber, daß es Turingmaschinen gibt (Beispiele lassen sich leicht konstruieren), die nach dieser Definition äquivalent sind, sich aber in ihren Programmen so unterscheiden, daß sie nicht durch die 4 Operationen ineinander überführt werden können. Man bekäme zwar zu dieser Äquivalenzrelation ein kleineres Repräsentantensystem als zu der in Definition 2 definierten Äquivalenzrelation, aber Versuche, für die Menge der Turingmaschinen bis zu 4 Befehlen ein solches Repräsentantensystem zu konstruieren, sind fehlgeschlagen. Ein Vorteil der Äquivalenzrelation nach Definition 2 ist der, daß zu jedem Repräsentanten, vorausgesetzt, daß das Halteproblem dafür gelöst ist, leicht mit Hilfe der 4 Operationen die Anzahl der zu diesem Repräsentanten äquivalenten Turingmaschinen ermittelt

werden kann.

Def. 3: $\mathcal{R}$ sei ein Repräsentantensystem zu der in Definition 2 definierten Äquivalenzrelation. $\mathcal{R}$ werde folgendermaßen in disjunkte Teilmengen zerlegt:

$$\mathcal{R} = \bigcup_{\substack{n \\ m \leq 2n}} (S(n,m) \cup U(n,m))$$

$$S(n,m) = S'(n,m) \cap \mathcal{R}, \quad U(n,m) = U'(n,m) \cap \mathcal{R}$$

$S'(n,m)$ und $U'(n,m)$ werden definiert durch:

$TM \in S'(n,m)$

1. TM ist haltende Turingmaschine
2. TM besitzt n Befehle
3. m Teilbefehle von TM werden beim Ablauf der Turingmaschine benutzt

$TM \in U'(n,m)$

1. TM ist nicht-haltende Turingmaschine
2. und 3. wie bei $S'(n,m)$

Die Teilmenge $\mathcal{R}_n$ von $\mathcal{R}$ enthalte alle Turingmaschinen aus $\mathcal{R}$ mit bis zu n Befehlen.

III Konstruktionsverfahren für eine Teilmenge eines Repräsentantensystems $\mathcal{R}$

In diesem Kapitel wird ein Konstruktionsverfahren hergeleitet zur Bestimmung der Teilmenge $\mathcal{R}_{n_o}$ von $\mathcal{R}$, wobei n_o folgende Bedeutung hat: Für jede Turingmaschine mit bis zu n_o Befehlen läßt sich das Halteproblem lösen. Die Aussage, daß das Halteproblem unlösbar ist, enthält keine Beschränkung über die Anzahl der Befehle und stellt somit zu dem oben Gesagten keinen Widerspruch dar. Praktisch durchgeführt wurde in dieser Arbeit die Bestimmung von $\mathcal{R}_4$.

Lemma 3: Sei $\mathcal{T}$ die Menge aller Turingmaschinen mit bis zu n_o Befehlen. Alle Turingmaschinen aus $\mathcal{T}$ werden mit Hilfe von Operation 1 aus Def. 2 so abgeändert, daß sie im ersten Teilbefehl des ersten Befehls (= Startbefehl) die Verschiebeoperation R haben. Die dadurch entstehende Untermenge von $\mathcal{T}$ sei $\mathcal{T}_1$. Für $\mathcal{T}_1$ gilt:

1) $\mathcal{T}_1$ enthält ein Teilrepräsentantensystem $\mathcal{R}_{n_o}$.
2) Für TM1, TM2 $\in \mathcal{T}_1$ und TM1 äquivalent zu TM2 gilt: Zur Überführung von TM1 in TM2 wird die Operation 1 nicht benötigt außer in dem trivialen Fall, daß in TM1 und TM2 der Startbefehl bereits Haltebefehl ist.

Beweis zu 1) klar

zu 2): Operation 3 und 4 lassen den Startbefehl unverändert. Wird Operation 1 benötigt, so ist durch Umstellung der Operationen sicherlich zu erreichen, daß sie nur einmal benötigt wird. Dann hätten aber TM1 und TM2 im Startbefehl unterschiedliche Verschiebeoperationen,

es sei denn, Operation 2 sei auf den Startbefehl angewandt worden, weil dieser bereits Haltebefehl ist.

Def. 4: Zu den nach Def. 1 zulässigen Teilbefehlen wird ein weiterer Teilbefehl eingeführt, der von folgender Form ist:

$$iZ* \qquad Z \in \{0,1\}$$

Dieser Teilbefehl kann auf zwei Arten interpretiert werden:

1) Er wird beim Ablauf der Turingmaschine nicht benutzt.

2) Soll beim Ablauf der Turingmaschine dieser Teilbefehl durchgeführt werden, so wird er als Haltebefehl ohne Schreibvorgang und Verschiebeoperation interpretiert.

Lemma 4: Die Untermenge $\mathcal{T}_2$ von $\mathcal{T}_1$ werde folgendermaßen gewonnen: Bei allen Turingmaschinen aus $\mathcal{T}_1$ wird jeder Teilbefehl, der beim Ablauf nicht benutzt wird, durch einen Teilbefehl der Form $iZ*$ ersetzt und Befehle, bei denen beide Teilbefehle beim Ablauf nicht benutzt werden, werden bei entsprechender Umnumerierung der übrigen Befehle ganz eliminiert. Diese Teilbefehle können effektiv gefunden werden, da für alle Turingmaschinen aus $\mathcal{T}_1$ das Halteproblem lösbar ist (vgl. hierzu Bemerkung zu Operation 4). Für $\mathcal{T}_2$ gilt dann:

1) $\mathcal{T}_2$ enthält ein Teilrepräsentantensystem $\mathcal{R}_{n_0}$.

2) Sei TM1, TM2 $\in \mathcal{T}_2$ und TM1 äquivalent zu TM2, so werden für die Überführung von TM1 in TM2 die Operationen 1 und 4 nicht benötigt.

Beweis zu 1) klar

zu 2): Daß die Operation 4 nicht benötigt wird, ist klar. In dem Fall, daß der Startbefehl Haltebefehl ist,

werden alle anderen Teilbefehle von der Form iZ* sein, d.h. Operation 1 und 2 sind für diese Turingmaschine identisch.

Bemerkung: Alle Turingmaschinen aus $\mathcal{T}_2$, die halten, haben genau einen expliziten Haltebefehl (d.h. nicht einen Teilbefehl der Form iZ*, der als Haltebefehl interpretiert werden kann), da Teilbefehle, die nicht benutzt werden, alle von der Form iZ* sind. Alle Turingmaschinen aus $\mathcal{T}_2$, die nicht halten, haben keinen expliziten Haltebefehl.

Lemma 5: Die Untermenge $\mathcal{T}_3$ von $\mathcal{T}_1$ entstehe dadurch, daß bei allen Turingmaschinen aus $\mathcal{T}_2$, die einen Haltebefehl haben, dieser die Verschiebeoperation R bekommt. Außerdem werde auf alle Turingmaschinen aus $\mathcal{T}_2$ die Operation 3 mit folgendem σ angewandt: Sei $i_1, i_2, \ldots$ die Folge der Befehlsnummern, so wie sie beim Ablauf der Turingmaschine hintereinander aufgerufen werden, so gibt es eine eindeutig bestimmte Permutation σ auf $1,\ldots,n$ (wenn n die Anzahl der Befehle ist), die die 1 festläßt und für die gilt:

Für alle $l>1$ und $\sigma(i_k)=l$ gibt es ein j, $1 \leq j<k$ mit $\sigma(i_j)=l-1$, d.h. läßt man in der permutierten Folge von jeder Befehlsnummer, die mehrmals auftaucht, nur jeweils das erste Exemplar stehen, bekommt man die Zahlen 1 bis n in natürlicher Reihenfolge. (Bemerkung: Ist die höchste Befehlsnummer in der Folge n, so handelt es sich um eine Turingmaschine mit n Befehlen, und jede Zahl, die kleiner als n ist, kommt ebenfalls in der Folge vor.)

Behauptung: $\mathcal{T}_3$ ist das gesuchte Teilrepräsentantensystem $\mathcal{R}_{n_o}$.

Beweis: Seien TM1, TM2 $\in \mathfrak{T}_3$ und TM1 äquivalent zu TM2. Operation 2 darf sicher nicht angewandt werden, da jede Turingmaschine von $\mathfrak{T}_3$ höchstens einen Haltebefehl hat, und dieser hat die Verschiebeoperation R. Da nur noch Operation 3 in Frage kommt, muß es ein σ geben, das TM1 in TM2 überführt. Ist σ aber von der Identität verschieden, wird es die Folge der Befehlsnummern so ändern, daß die übrigbleibenden n Zahlen 1 bis n nach Streichung der Wiederholung nicht mehr in natürlicher Reihenfolge sind.

Beispiel: n=7, Folge: 1,2,5,3,5,4,2,7,5,6,3,...
so ist $\sigma = (1)(5,3,4)(6,7)$ und σ auf die Folge angewandt, liefert: 1,2,3,4,3,5,2,6,5,7,3,...

Lemma 6: $S(n,m)=U(n,m)=\emptyset$ für $m < n$
also $\mathfrak{R}_{n_o} = \bigcup_{\substack{n=1 \\ 2n \geq m > n}}^{n_o} (S(n,m) \cup U(n,m))$

Beweis: klar wegen Lemma 4.

Konstruktionsverfahren

Zunächst kann man aus den Lemmata 3, 4 und 5 ein hinreichendes Kriterium dafür angeben, wann eine Turingmaschine aus $\mathfrak{R}_{n_o}$ ist. Dies ist der Fall, wenn gilt:

1. Der Startbefehl hat die Verschiebeoperation R.
2. Alle beim Ablauf nicht benutzten Teilbefehle haben die Form iZ*.
3. Wenn bei einem Befehl beide Teilbefehle von der Form iZ* sind, so auch bei allen folgenden.
4. Der Haltebefehl (es gibt höchstens einen expliziten)

hat die Verschiebeoperation R.

5. Streicht man in der Folge der Befehlsnummern, wie sie beim Ablauf auftreten, alle Wiederholungen, so sind die übrigbleibenden Zahlen in natürlicher Reihenfolge.

Die Mengen S(n,m) und U(n,m) werden nun sukzessive von S(1,1) beginnend aufgebaut, wobei jeweils bei den haltenden Turingmaschinen, also bei denen aus S(n,m), der Haltebefehl nacheinander durch alle möglichen Nicht-Haltebefehle ersetzt werden, die die obigen 5 Bedingungen nicht verletzen. Dazu denke man sich bei jeder Turingmaschine den Befehlsspeicher auf n Befehle verlängert, und alle nicht benutzten Plätze für Teilbefehle sollen mit Teilbefehlen der Form iZ* besetzt werden. Im folgenden werden bei der Beschreibung von Turingmaschinen die Befehle nur bis zum letzten relevanten Befehl aufgeführt, d.h. bis zum letzten der Befehle, die mindestens einen Teilbefehl besitzen, der nicht von der Form iZ* ist.

Wegen Bedingung 1 bestehen S(1,1) und U(1,1) aus je zwei Turingmaschinen:

$$S(1,1) = \left\{\begin{pmatrix}100R0\\11*\end{pmatrix}, \begin{pmatrix}101R0\\11*\end{pmatrix}\right\}$$

$$U(1,1) = \left\{\begin{pmatrix}100R1\\11*\end{pmatrix}, \begin{pmatrix}101R1\\11*\end{pmatrix}\right\}$$

Lemma 6a: Seien nun die Mengen $S(i,j)$, $i \leq n$, $j \leq m$ $(i,j) \neq (n,m)$ bereits bestimmt, so wird $S(n,m)$ mit Hilfe folgender Mengen bestimmt:

$$S_{n,m} = S(n,m-1) \qquad \text{falls } 2n = m$$

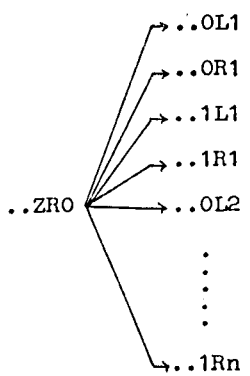
$$S_{n,m} = S(n,m-1) \cup S(n-1,m-1) \qquad \text{falls } n < m < 2n$$

$$S_{n,m} = S(n-1,m-1) \qquad \text{falls } n = m$$

Zu jeder Turingmaschine TM aus $S_{n,m}$ gibt es eine weitere Turingmaschine aus $S_{n,m}$, die sich nur im Haltebefehl von TM unterscheidet, und zwar wird einmal eine "1" und einmal eine "0" geschrieben. Von jedem solchen Paar darf bei der Variation des Haltebefehls jeweils nur eine Turingmaschine genommen werden. Es wird also genau die Hälfte aller Turingmaschinen aus $S_{n,m}$ dazu benötigt, um S(n,m) aufzubauen. Es müssen zwei Fälle unterschieden werden:

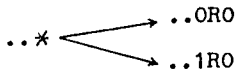
1) $TM \in S(n,m-1)$

Durch Variation des Haltebefehls entstehen 4·n neue Turingmaschinen, und zwar werden nacheinander für den Haltebefehl folgende Teilbefehle eingesetzt:



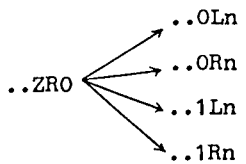
Von den 5 Bedingungen her gibt es keinerlei Einschränkung. Bei den Nummern der aufzurufenden Befehle darf keine Zahl größer als n auftauchen, da nur Turingmaschinen aus S(n,m) gesucht werden. Bei diesen 4·n Turingmaschinen wird der Ablauf so lange verfolgt, bis ihr Nichthalten festgestellt wird oder bis ein Teilbefehl der Form iZ* durchgeführt werden soll (explizite Haltebefehle enthalten diese 4·n Turing-

maschinen nicht mehr). Dieser wird dann zum Haltebefehl gemacht, wobei zwei Turingmaschinen entstehen (R wegen Bedingung 4):



2) TM $\in$ S(n-1,m-1)

Bei dieser Turingmaschine entstehen durch Variation des Haltebefehls 4 Turingmaschinen:

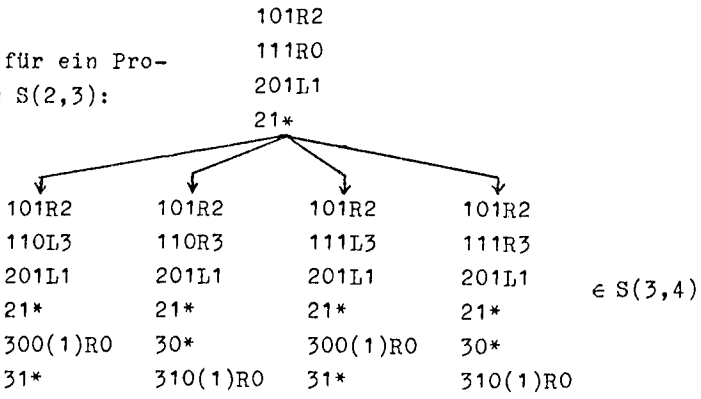


Die Nummer des aufzurufenden Befehls muß n sein, da TM nur n-1 echte Befehle hatte und Turingmaschinen mit n Befehlen gesucht sind. Diese 4 Turingmaschinen machen beim Ablauf nur einen Schritt mehr, da sie nach Variation des Haltebefehls von TM sofort auf einen Teilbefehl der Form 1Z* laufen, der analog zum Fall 1) abgeändert wird. Alle 4 Turingmaschinen halten.

Die Menge U(n,m) besteht aus den Turingmaschinen, die nach Variation des Haltebefehls von S(n,m) nicht halten. Bei der Ermittlung von S(n,m) fällt also U(n,m-1) mit an. In Abbildung 1 ist für ein Beispiel das Konstruktionsverfahren erläutert. In Abbildung 2 ist schematisch noch einmal die Gesamtkonstruktion wiedergegeben.

Bei diesem Konstruktionsverfahren werden die 5 Bedingungen nicht verletzt. Da sonst bei der Variation der Teilbefehle (jeder Teilbefehl wird im Laufe des Verfahrens genau einmal

Beispiel für ein Programm aus S(2,3):



Beispiel für ein Programm aus S(3,3):

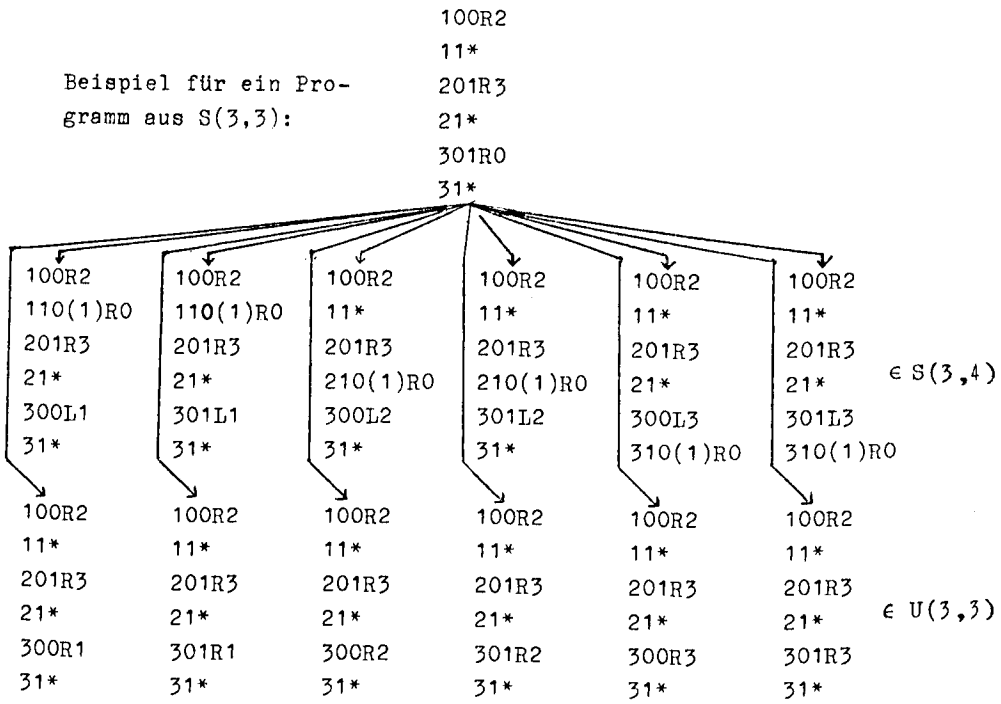


Abb. 1 Konstruktion der Mengen S(3,4)und U(3,3)

$S_{n,m} = S(3,3) \cup S(2,3)$

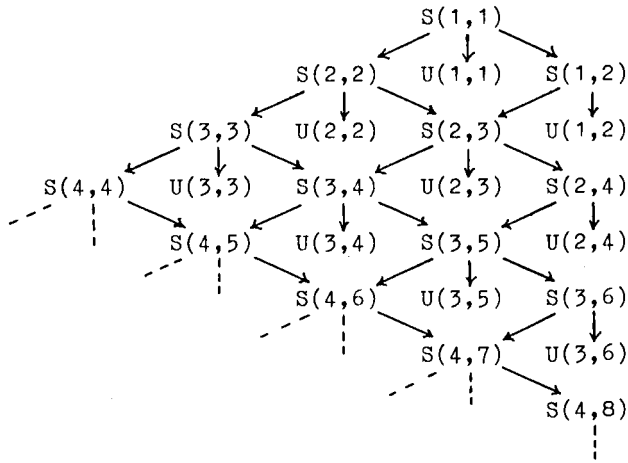


Abb. 2

durchvariiert) keine Einschränkung besteht und außerdem dieselbe Turingmaschine nicht mehrmals entstehen kann, wird hiermit genau das Repräsentantensystem $\mathcal{R}_{n_o}$ konstruiert.

Bemerkung: Für die Lösung des Busy-Beaver-Problems könnte man die Äquivalenzrelation und damit die 5 Bedingungen dahingehend erweitern, daß man im Startbefehl als zu schreibendes Zeichen nur die 1 zuläßt (siehe [5]). Das wurde aber bewußt vermieden, um einen Überblick über alle Turingmaschinen aus $\mathcal{R}_{n_o}$ zu bekommen.

IV Hinreichende Kriterien für das Nichthalten von Turingmaschinen

Im folgenden werden vier Kriterien aufgestellt, bei deren Erfülltsein das Nichthalten einer Turingmaschine nachgewiesen ist. Die ersten beiden wurden auch von S. Lin und T. Rado [5] verwandt.

Def. 5: Der Ablauf einer Turingmaschine erfolgt in Schritten (ein Schritt entspricht der Durchführung eines Teilbefehls). Vor jedem Schritt wird der Zustand der Turingmaschine durch das Tripel (i_j,a_j,b_j) beschrieben. Dabei sei:

- i_j die Nummer des Befehls, der im j-ten Schritt durchgeführt wird.
- a_j die Nummer des Arbeitsfeldes für den j-ten Schritt.
- b_j eine Belegungsfunktion der Art, daß $b_j(k)$ die Belegung des Feldes k vor Ausführung des j-ten Schrittes darstellt $(b_j(k) \in \{0,1\})$.

Kriterium 1: Für zwei natürliche Zahlen m und n mit $n > m$ und eine Turingmaschine TM mit den Zuständen $\{(i_j,a_j,b_j)\}_{j=1,2,\ldots}$ werde definiert, wobei o.B.d.A. $a_n \geq a_m$ sein soll:

$$l_{nm} = \min \{a_j \mid j \in \{m,\ldots,n\}\}$$

$$r_{nm} = \max \{a_k \mid k \in \{1,\ldots,n\}\}$$

Gilt nun für m und n:

1) $i_m = i_n$

2) $b_m(k) = b_n(k+(a_n-a_m))$; $k \in \{l_{nm},\ldots,r_{nm}\}$

so hält TM nie an.

Bemerkung: 2) ist identisch mit:

$$2') \quad b_m(k) = b_n(k+(a_n-a_m)) \; ; \quad k \geq l_{nm}$$

da die Felder rechts von r_{nm} bis zum Zustand (i_n,a_n,b_n) noch nicht als Arbeitsfelder aufgetreten sind und damit alle die Belegung 0 haben.

Für den Fall $a_n < a_m$ läßt sich das Kriterium analog formulieren.

Beweis: Numeriert man beim Zustand (i_n,a_n,b_n) die Felder des Bandes um, was sicher keinen Einfluß auf die Frage nach dem Nichthalten von TM hat, mit Hilfe folgender Zuordnung:

$$f(j) = j-(a_n-a_m)$$

und ändert entsprechend das Tripel (i_n,a_n,b_n) ab in (i_n^*,a_n^*,b_n^*) mit:

$$i_n = i_m = i_n^* \; ; \; a_n^* = a_m \; ; \; b_n^*(j) = b_n(j-(a_n-a_m)) = b_m(j), \; j \geq l_{nm}$$

so unterscheiden sich b_m und b_n^* höchstens in Feldern, die links von l_{nm} liegen, also in Feldern, die beim Ablauf vom Zustand (i_m,a_m,b_m) zum Zustand (i_n,a_n,b_n) nicht als Arbeitsfelder auftreten. Man kann also für die nächsten n-m Schritte (i_n^*,a_n^*,b_n^*) mit (i_m,a_m,b_m) identifizieren. Nach Ausführung der nächsten n-m Schritte wird dann aber wieder der Zustand (i_n,a_n,b_n) erreicht. Da sich dies beliebig wiederholt, hält TM nie an.

Bemerkung: Kriterium 1 erkennt einfache Schleifen (im Gegensatz zu sich vergrößernden Schleifen bei Kriterium 4), die sich beliebig wiederholen. Die Schleifen können auf dem Band nach rechts ($a_n > a_m$) oder nach Links ($a_n < a_m$) wandern oder auch an derselben Stelle ($a_n = a_m$) bleiben.

Kriterium 2: Turingmaschinen, bei denen in keinem Teilbefehl an 5. Stelle (dort steht die Nummer des als nächsten durchzuführenden Befehls) die Nummer eines Befehls steht, der einen Haltebefehl oder einen Befehl der Form $iZ*$ als Teilbefehl besitzt, halten trivialerweise nie.

Kriterium 3: Ein möglicher Zustand der Turingmaschine TM sei (i_r,a_r,b_r), wobei eventuell von b_r nur $b_r(a_r)$ bekannt ist. (Dies muß bekannt sein, um entscheiden zu können, welcher Teilbefehl vom Befehl i_r als nächster Schritt durchgeführt wird.) Man kann sich nun überlegen, welche Möglichkeiten für den Zustand $(i_{r-1},a_{r-1},b_{r-1})$ in Frage kommen. Der Befehl i_{r-1} muß einen Teilbefehl der Form

$$i_{r-1}Z_1Z_2V\ i_r \qquad (*)$$

besitzen. Für diesen Teilbefehl und den Zustand $(i_{r-1},a_{r-1},b_{r-1})$ muß gelten, wenn V=R ist:

a) $a_{r-1} = a_r - 1$ wegen V=R

b) $Z_2 = b_r(a_{r-1})$, da Z_2 nach Durchführung des (r-1)-ten Schrittes die Belegung von a_{r-1} ist.

c) $b_{r-1}(a_{r-1}) = Z_1$, damit auch der Teilbefehl $(*)$ von i_{r-1} durchgeführt wird.

d) $b_r(k) = b_{r-1}(k)$ für $k \neq a_{r-1}$, da sich die Belegungen der Felder $k \neq a_{r-1}$ im (r-1)-ten Schritt nicht ändern.

Ist V = L, muß nur a) in a') geändert werden:

a') $a_{r-1} = a_r + 1$

Wäre TM eine haltende Turingmaschine, müßte sie einmal einen Zustand (i_r,a_r,b_r) erreichen, für den gilt:

Ist $b_r(a_r) = Z$, so ist der Teilbefehl $i_rZ\ldots$ ein Haltebefehl oder ein Befehl der Form i_rZ*.

Man stellt nun zunächst alle möglichen Zustände auf, die im nächsten Schritt zum Halt führen. Die Gesamtheit dieser Zustände heiße Zustände der 1. Stufe. Hat man nun die Zustände der n-ten Stufe, so kann man nach obigem Verfahren zu jedem dieser Zustände alle möglichen vorhergehenden Zustände ermitteln. Die Gesamtheit der Zustände, die sich dabei ergeben, seien die Zustände der (n+1)-ten Stufe. Es kann nun eintreten, daß es zu einem Zustand keinen möglichen vorhergehenden Zustand gibt. Der Zustand (i_j,a_j,b_j) hat zum Beispiel keinen vorhergehenden Zustand, wenn für alle Befehle der Form $i_{j-1}Z_1Z_2Ri_j$ gilt $Z_2 \neq b_j(a_j-1)$ bzw. für alle Befehle der Form $i_{j-1}Z_1Z_2Li_j$ gilt $Z_2 \neq b_j(a_j+1)$. Tritt es nun ein, daß es keinen Zustand der n-ten Stufe gibt, wobei auf keiner früheren Stufe ein Zustand existierte, der als Befehl den Befehl 1 hatte und bei dem alle Felder mit 0 belegt waren (dies würde einem normalen Start von TM entsprechen, der dann bei (i_r,a_r,b_r) zu einem Halt führte), hält TM nie an, da auf keinem Weg ein Haltebefehl oder ein Befehl der Form $iZ*$ erreicht werden kann. In Abbildung 3 ist die Wirkungsweise dieses Kriteriums demonstriert.

Kriterium 4: Die Turingmaschine TM nehme beim Ablauf folgende 4 Zustände an:

$$(i_k,a_k,b_k),\quad (i_l,a_l,b_l),\quad (i_m,a_m,b_m),\quad (i_p,a_p,b_p)$$

wobei $k < l < m < p$ sein soll.

Zunächst soll mit Worten eine Übersicht über das Kriterium gegeben werden. Die 4 Zustände stellen eine Art Wendepunkte der

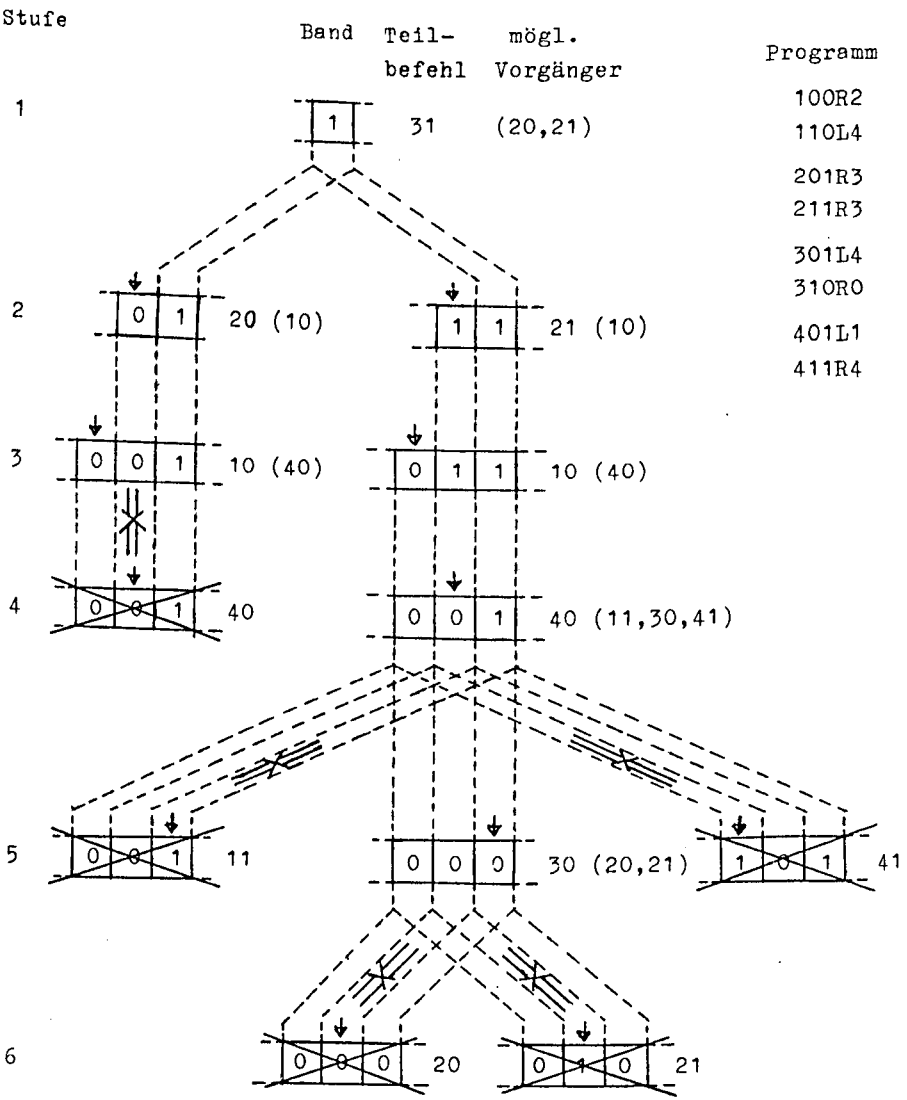


Abb. 3: Beispiel für Kriterium 3

Bewegungen des Lese- und Schreibkopfes auf dem Band dar. O.B.d.A. soll der erste Zustand ein linker Wendepunkt sein, d.h. die auf diesen Zustand folgenden Zustände haben Arbeitsfelder rechts von a_k. Der dritte Zustand stellt dann wieder einen linken Wendepunkt dar, wobei $a_k \leq a_m$ oder $a_k \geq a_m$ sein kann. Die Zustände 2 und 4 sind rechte Wendepunkte. Die Belegung b_m ist in gewisser Hinsicht eine Wiederholung der Belegung b_k. Schiebt man nämlich bei der Belegung b_k ein Stück Band mit einer bestimmten Belegung ein und numeriert die Felder um, so kann man b_k mit b_m identifizieren mit Ausnahme solcher Felder, die beim weiteren Ablauf nicht mehr als Arbeitsfelder auftreten. Die Belegung von b_k und damit auch die von b_m ist folgendermaßen aufgebaut: von links nach rechts, also von a_k (linker Wendepunkt) bis a_1 (rechter Wendepunkt) kommt zunächst ein Anfangsstück, dann eine Anzahl von gleichlangen Stücken (im weiteren Symmetrie-Elemente genannt), deren Belegungen identisch sind, und ein Endstück. Die Belegung b_m hat neben dem gleichen Anfangs- und Endstück genau ein Symmetrieelement mehr. Da außerdem $i_k=i_m$ sein soll, handelt es sich also hier um Schleifen, die sich im Gegensatz zu Kriterium 1 vergrößern (um jeweils ein Symmetrieelement). Das bisher Gesagte gilt analog für die beiden rechten Wendepunkte, soll aber hier und auch in der nachfolgenden exakten Formulierung des Kriteriums nicht mit aufgeführt werden.

Es werden nun Kriterien aufgestellt, bei deren Erfülltsein folgendes nachgewiesen ist:

Hat TM einen Zustand (i_r,a_r,b_r) erreicht, wobei b_r neben dem Anfangsstück und Endstück von b_k (sowie von b_m) n Symmetrieelemente besitzt, so nimmt TM beim weiteren Ablauf einen Zustand (i_s,a_s,b_s) ein, bei dem b_s neben demselben Anfangs- und Endstück

genau n+1 Symmetrieelemente besitzt. Beide Zustände stellen wieder linke Wendepunkte dar und es gilt $i_r=i_s=i_k=i_m$. Außerdem ist der Abstand von a_r zu a_s derselbe wie von a_k zu a_m.

Ist $a_k < a_m$, so ist beim Zustand (i_k,a_k,b_k) die Belegung links von a_k beliebig und für den weiteren Ablauf ohne Bedeutung. Ist $a_k > a_m$, darf kein Feld links von a_k bis zum Erreichen des Zustandes (i_k,a_k,b_k) als Arbeitsfeld aufgetreten sein, womit alle Felder links von a_k die Belegung 0 haben. Damit sind b_k und b_s nach dem Einschub eines Symmetrieelementes bei b_k und entsprechender Umnumerierung aller Felder, die noch im weiteren Ablauf als Arbeitsfelder benutzt werden, identisch.

Der Beweis, daß zum Zustand (i_r,a_r,b_r) mit n Symmetrieelementen der Zustand (i_s,a_s,b_s) mit n+1 Symmetrieelementen existiert, wird folgendermaßen geführt: Man weist nach, daß die Anzahl der Symmetrieelemente, einmal ausgehend von einem linken und einmal von einem rechten Wendepunkt, insofern nur einen unwesentlichen Einfluß auf den weiteren Ablauf hat, als durch den Einschub eines zusätzlichen Symmetrieelementes bei einem linken Wendepunkt anschließend beim nachfolgenden rechten Wendepunkt ebenfalls ein zusätzliches Symmetrieelement auftaucht und umgekehrt. Ist alles bisher Geschilderte für die Turingmaschine TM erfüllt, hält TM nicht an.

Formulierung des Kriteriums:

Für die Zustände (i_k,a_k,b_k), (i_l,a_l,b_l), (i_m,a_m,b_m), (i_p,a_p,b_p) muß folgendes gelten (nur für linke Wendepunkte formuliert):

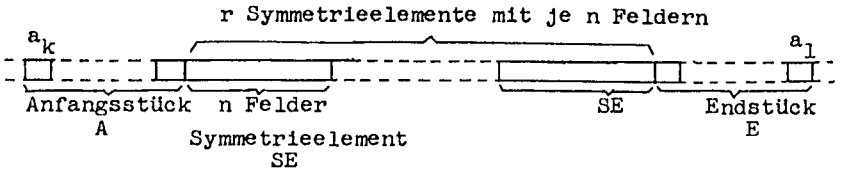
1) $i_k = i_m$

2) o.B.d.A. $a_k < a_l$

3) Ist $a_k > a_m$, so gilt: $\min \{a_j \mid j=1,\ldots,k\} = a_k$

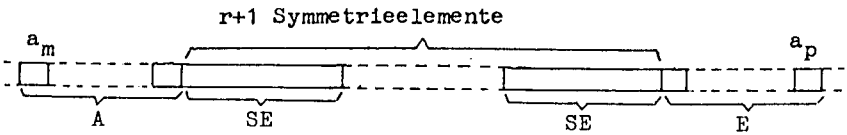
$\min \{a_j \mid j=1,\ldots,m\} = a_m$

4) Ist $n=a_p-a_m-(a_1-a_k)$, so läßt sich die Belegung b_k folgendermaßen aufspalten:



Die Belegung der r Symmetrieelemente ist identisch

Für die Belegung b_m gilt:



5) Sei n_1 das Anfangs- und n_2 das Endfeld irgendeines Symmetrieelementes der Belegung b_k. Es gibt zwei Zustände (i_r,a_r,b_r) und (i_s,a_s,b_s), die in dieser Reihenfolge im weiteren Ablauf auf (i_k,a_k,b_k) folgen, für die gilt:

i) $i_r = i_s$

ii) $a_r = n_1$; $a_s = n_2+1$

iii) Für $k \leq j < r$ gilt: $a_j < a_r$

Für $r \leq j < s$ gilt: $a_r \leq a_j < a_s$

Sind die Punkte 1) bis 5) erfüllt, so spielt die Anzahl der Symmetrieelemente bei b_k keine Rolle mehr, da für die Symmetrieelemente ab dem Feld n_1 gilt:

Jedes Symmetrieelement wird vom Lese- und Schreibkopf nur einmal (von links) erreicht und zwar mit Durchführung desselben Befehls,

so daß sie bei Erreichen des Zustandes (i_1,a_1,b_1) alle dieselbe Belegung haben.

Verläßt der Lese- und Schreibkopf ein Symmetrieelement nach Erreichen desselben noch einmal nach links, so kann eventuell das Ersetzen des Punktes 5) durch 5') zum Ziele führen.

5') n_1 sei das Anfangs- und n_2 das Endfeld des ersten, A unmittelbar benachbarten, Symmetrieelementes der Belegung b_k. Es gibt 6 Zustände $(i_{j_r},a_{j_r},b_{j_r})$ $r=1,\ldots,6$, die TM während des Ablaufs einnimmt, und für diese gilt:

i) $k<j_1<j_2<j_3<j_4<j_5<j_6<1$

ii) $a_{j_1} = n_1 \quad ; a_k \leq a_s < n_1 \qquad$ für $k \leq s < j_1$

$a_{j_2} = n_1-1 \; ; n_1 \leq a_s \leq n_2 \qquad$ für $j_1 \leq s < j_2$

$a_{j_3} = n_1 \quad ; a_k \leq a_s < n_1 \qquad$ für $j_2 \leq s < j_3$

$a_{j_4} = n_2+1 \; ; n_1 \leq a_s \leq n_2 \qquad$ für $j_3 \leq s < j_4$

$a_{j_5} = n_2 \quad ; n_2 < a_s \qquad$ für $j_4 \leq s < j_5$

$a_{j_6} = n_1-1 \; ; n_1 \leq a_s \leq n_2 \qquad$ für $j_5 \leq s < j_6$

iii) $i_{j_1} = i_{j_3} = i_{j_4}$

$i_{j_2} = i_{j_5} = i_{j_6}$

iv) $b_{j_1}(n_1+s) = b_{j_6}(n_1+s)$

$b_{j_2}(n_1+s) = b_{j_3}(n_1+s) \quad$ (trivial)

$b_{j_4}(n_1+s) = b_{j_5}(n_1+s) \quad$ (trivial)

} für $s = 0,\ldots,n_2-n_1$

$b_{j_1}(a_k+s) = b_{j_3}(a_k+s) \qquad$ für $s = 0,\ldots,n_1-a_k-1$

Bandbelegung	Befehl	
a_k $n_1 n_2$ a_1		
. 0 0 | 0 0 | 0 0 | . . | 0 0 | 1 0 0 . (A SE SE SE E)	1	linker Wendepunkt
n Symmetrieel.		
. 1 0 | 0 0 | 0 0 | . . | 0 0 | 1 0 0 .	2	
. 1 0 | 0 0 | 0 0 | . . | 0 0 | 1 0 0 .	2	
. 0 0 | 0 0 | 0 0 | . . | 0 0 | 1 0 0 .	3	
. 0 1 | 0 0 | 0 0 | . . | 0 0 | 1 0 0 .	4	$a_{j_1} = n_1$; $i_{j_1} = 4$
. 0 1 | 1 0 | 0 0 | . . | 0 0 | 1 0 0 .	1	$a_{j_2} = n_1 - 1$; $i_{j_2} = 1$
. 0 0 | 1 0 | 0 0 | . . | 0 0 | 1 0 0 .	1	
. 1 0 | 1 0 | 0 0 | . . | 0 0 | 1 0 0 .	2	
. 1 0 | 1 0 | 0 0 | . . | 0 0 | 1 0 0 .	2	
. 0 0 | 1 0 | 0 0 | . . | 0 0 | 1 0 0 .	3	
. 0 1 | 1 0 | 0 0 | . . | 0 0 | 1 0 0 .	4	$a_{j_3} = n_1$; $i_{j_3} = 4$
. 0 1 | 1 0 | 0 0 | . . | 0 0 | 1 0 0 .	3	
. 0 1 | 1 1 | 0 0 | . . | 0 0 | 1 0 0 .	4	$a_{j_4} = n_2+1$; $i_{j_4} = 4$
. 0 1 | 1 1 | 1 0 | . . | 0 0 | 1 0 0 .	1	$a_{j_5} = n_2$; $i_{j_5} = 1$
. 0 1 | 1 0 | 1 0 | . . | 0 0 | 1 0 0 .	1	
. 0 1 | 0 0 | 1 0 | . . | 0 0 | 1 0 0 .	1	$a_{j_6} = n_6$; $i_{j_6} = 1$
. 0 0 | 0 0 | 1 0 | . . | 0 0 | 1 0 0 .	1	
. 0 1 | 1 1 | 1 1 | . . | 1 1 | 1 0 0 .	4	
. 0 1 | 1 1 | 1 1 | . . | 1 1 | 1 0 0 .	3	
. 0 1 | 1 1 | 1 1 | . . | 1 1 | 1 1 0 .	4	rechter Wendepunkt

Programm:

Befehl:	1	2	3	4
	101R2	200L2	301R4	401L1
	110L1	210R3	310R0	411R3

Abb. 4

Ist 5') erfüllt, nimmt jedes Symmetrieelement der Belegung b_k beim Ablauf von TM von (i_k, a_k, b_k) drei sich immer wiederholende Belegungen an:

1. Ausgangsbelegung;
2. Zwischenbelegung für den Fall, daß der Lese- und Schreibkopf das Symmetrieelement erreicht und nach links wieder verlassen hat;
3. Zwischenbelegung für den Fall, daß der Lese- und Schreibkopf sich rechts vom Symmetrieelement befindet.

Wird das Symmetrieelement von rechts erreicht, verläßt der Lese- und Schreibkopf es nach links und die Ausgangsbelegung wird wieder hergestellt. Befindet sich der Lese- und Schreibkopf nicht im Anfangsstück A, ist die Belegung von A immer dieselbe. Da alle Änderungen der Belegung des Anfangsstücks sowie aller Symmetrieelemente mit derselben Befehlsfolge geschehen, hat jedes Symmetrieelement, unabhängig von der Anzahl der vorhandenen Symmetrieelemente, wenn der Lese- und Schreibkopf das Endstück E erreicht, dieselbe Belegung. Abbildung 4 verdeutlicht Punkt 5') an einem Beispiel.

Hat man von TM gezeigt, daß dieses Kriterium sowohl für die linken Wendepunkte als auch (analog umformuliert) für die rechten Wendepunkte erfüllt ist, so ist nachgewiesen, daß TM nicht hält.

V Ergebnisse über haltende Turingmaschinen mit bis zu 5 Befehlen

Alle Aussagen und Statistiken dieses Kapitels beziehen sich auf das Repräsentantensystem $\mathcal{R}$. Um die Hauptergebnisse formulieren zu können, müssen einige Funktionen definiert werden:

Def. 6: Die Funktionen $\sigma, \varphi, \beta : S(n,m) \to \mathbb{N}$ für alle $n,m \in \mathbb{N}$ $n \leq m \leq 2n$ und $k,s,f,b: \mathbb{N}^2 \to \mathbb{N}$ werden definiert durch:

$\sigma(TM)$ = Anzahl der Teilbefehle, die bis zum Haltebefehl einschließlich durchgeführt werden (Laufzeit);

$\varphi(TM)$ = Anzahl der verschiedenen Felder, die beim Ablauf als Arbeitsfelder auftreten (Feldweite von TM);

$\beta(TM)$ = Anzahl der Einsen, die nach dem Halt auf dem Band stehen;

$k(n,m) = |S(n,m)|$ = Anzahl der Turingmaschinen in $S(n,m)$

$s(n,m) = \max \{ \sigma(TM) \mid TM \in S(n,m) \}$

$f(n,m) = \max \{ \varphi(TM) \mid TM \in S(n,m) \}$

$b(n,m) = \max \{ \beta(TM) \mid TM \in S(n,m) \}$ (Busy-Beaver-Spiel)

Def. 7: Um einen noch genaueren Überblick über das Repräsentantensystem $\mathcal{R}$ zu bekommen, sollen die Mengen $S(n,m)$ noch in disjunkte Teilmengen zerlegt werden:

$$S(n,m) = S_{OR}(n,m) \cup S_{1R}(n,m) \cup S_{OB}(n,m) \cup S_{1B}(n,m)$$

Die einzelnen Teilmengen werden definiert durch:

$TM \in S_{OR}(n,m) \iff$ 1. $TM \in S(n,m)$

2. Der Startbefehl lautet: 100R2

3. Alle beim Ablauf benötigten Arbeitsfelder haben eine Nummer ≥ 0, d.h., nur die "rechte Seite" des Bandes wird benutzt.

n,m	k_{CR}	s_{OR}	f_{OR}	b_{OR}	k_{1R}	s_{1R}	f_{1R}	b_{1R}	k_{OB}	s_{OB}	f_{OB}	b_{OB}	k_{1B}	s_{1B}	f_{1B}	b_{1B}
2,2	2	2	2	1	2	2	2	2	--	--	--	--	--	--	--	--
2,3	2	4	2	1	8	3	2	2	--	--	--	--	--	--	--	--
2,4	2	6	3	2	8	4	2	2	--	--	--	--	12	6	4	4
3,3	8	3	3	2	8	3	3	3	--	--	--	--	--	--	--	--
3,4	38	5	3	2	54	5	3	3	4	5	3	2	36	5	4	4
3,5	138	9	4	4	266	10	4	4	32	9	4	4	282	17	5	5
3,6	378	14	5	4	590	13	5	4	180	17	6	5	1492	21	7	6
4,4	24	4	4	3	24	4	4	4	8	4	3	3	8	4	3	3
4,5	320	7	4	4	424	7	4	4	98	8	4	4	266	8	5	5
4,6	2376	18	5	5	3362	12	5	5	1210	18	6	5	4704	20	6	6
4,7	14148	24	7	6	20446	23	7	5	10864	30	8	7	45548	38	10	9
4,8	45504	43	11	7	59664	58	10	7	56082	71	12	10	230762	107	16	13

Tabelle 1

n,m	k_{OR}	s_{OR}	f_{OR}	b_{OR}	k_{1R}	s_{1R}	f_{1R}	b_{1R}
5,5	96	5	5	4	96	5	5	5
5,6	2036	9	5	5	2320	9	5	5
5,7	27804	19	6	6	36180	19	6	6
5,8	276610	33	9	8	357192	39	9	8
5,9	?	178	16	12	?	84	14	13
5,10	?	237	21	18	?	189	22	19

Tabelle 2

$TM \in S_{1R}(n,m) \iff$ 1. und 3. wie bei $S_{0R}(n,m)$
2. Der Startbefehl lautet: 101R2

$TM \in S_{0B}(n,m) \iff$ 1. und 2. wie bei $S_{0R}(n,m)$
3. "Beide Seiten" des Bandes werden benutzt.

$TM \in S_{1B}(n,m) \iff$ 1. und 3. wie bei $S_{0B}(n,m)$
2. wie bei $S_{1R}(n,m)$

Bemerkung: Der erste Index bei den Teilmengen gibt an, ob im Startbefehl 0 oder 1 geschrieben wird, der zweite, ob die Turingmaschine nur die rechte Seite (R) oder beide Seiten (B) des Bandes benutzt.

Def. 8: Entsprechend den Teilmengen von S(n,m) sollen auch die Funktionen der Def. 6 indiziert werden. Dabei bedeuten die indizierten Funktionen die Einschränkung der Ausgangsfunktionen auf die Teilmenge mit dem gleichen Index.

Tabelle 1 gibt alle Funktionswerte der in den Definitionen 6 und 8 definierten Funktionen bis (n,m) = (4,8) wieder. Da $\mathcal{R}_4$ vollständig bestimmt wurde, sind alle Werte, die Turingmaschinen mit bis zu 4 Befehlen betreffen, als bewiesen anzusehen. Über die Bedeutung von Tabelle 2, die hier nur aus Gründen der Vollständigkeit mit aufgeführt ist, gibt Kapitel VI Auskunft.

Es folgen nun Statistiken über Laufzeit (σ), Zahl der Einsen (β) und Feldweite (φ). In den Tabellen stehen jeweils die Anzahlen der Turingmaschinen in den einzelnen Teilmengen mit gleicher Laufzeit bzw. gleicher Anzahl der Einsen bzw. gleicher Feldweite.

Statistik über Laufzeiten

$S_{1B}(4,8)$

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	-	-	-	9600	24144
1	36668	35382	30880	23002	17886	13396	9640	6780	5388	3796
2	2826	2316	1800	1380	1082	928	622	490	388	374
3	286	220	228	198	174	114	86	64	80	68
4	54	38	38	34	32	22	26	20	14	22
5	14	20	22	6	4	8	10	10	2	6
6	6	6	8	6	8	2	2	2	8	4
7	2	-	-	-	2	2	-	-	4	-
8	-	-	-	2	2	-	-	-	-	-
9	-	-	-	-	-	-	4	2	-	-
10	-	-	-	-	-	-	-	2	-	-

$S_{0B}(4,8)$

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	-	-	-	1344	3408
1	8284	7596	8018	5802	5016	3738	2896	2094	1688	1410
2	1000	756	658	456	382	274	262	170	162	100
3	100	76	48	44	42	42	24	14	12	30
4	18	8	16	16	10	10	4	4	10	4
5	2	-	4	6	2	2	4	2	-	-
6	-	-	-	-	2	2	-	2	2	4
7	-	2	-	-	-	-	-	-	-	-

$S_{1R}(4,8)$

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	-	-	-	9696	14064
1	12776	8530	5376	3748	1898	1358	846	446	340	212
2	132	78	46	22	28	22	16	2	12	2
3	-	6	-	-	-	2	-	-	-	2
4	-	-	2	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	2	-

$S_{0R}(4,8)$

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	-	-	-	3456	7200
1	10200	7794	5422	3738	2550	1642	1190	570	486	292
2	262	142	168	94	72	40	50	36	24	6
3	14	8	8	8	4	12	2	-	4	2
4	2	2	2	2	-	-	-	-	-	-

$S_{1B}(4,7)$

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	-	-	4608	10224	11192
1	7634	4684	2618	1748	1094	538	324	196	172	164
2	114	80	38	30	18	12	12	6	6	8
3	12	2	6	4	2	-	-	-	2	-

$S_{0B}(4,7)$

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	-	-	576	2160	2508
1	1962	1328	912	446	394	222	120	70	58	40
2	26	18	8	-	-	10	2	2	-	-
3	2	-	-	-	-	-	-	-	-	-

		0	1	2	3	4	5	6	7	8	9
$S_{1R}(4,7)$	0	-	-	-	-	-	-	-	7392	5592	4132
	1	1586	840	456	252	78	72	20	12	6	4
	2	2	-	-	2	-	-	-	-	-	-
$S_{0R}(4,7)$	0	-	-	-	-	-	-	-	2688	3744	3876
	1	1734	1000	564	246	116	90	26	14	10	18
	2	10	4	4	-	4	-	-	-	-	-

	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
$S_{1B}(4,6)$	-	1392	1604	1074	330	218	38	22	6	4	-	-	4	10	-	2	-
$S_{0B}(4,6)$	-	288	356	382	70	86	8	16	2	-	-	-	-	2	-	-	-
$S_{1R}(4,6)$	-	1776	1080	372	84	12	28	10	-	-	-	-	-	-	-	-	-
$S_{0R}(4,6)$	-	912	816	524	70	36	16	-	-	-	-	-	-	2	-	-	-
$S_{1B}(3,6)$	-	144	304	394	290	150	74	54	34	16	8	4	2	8	4	4	2
$S_{0B}(3,6)$	-	-	16	42	36	34	18	8	8	10	2	4	2	-	-	-	-
$S_{1R}(3,6)$	-	192	204	102	48	22	18	2	2	-	-	-	-	-	-	-	-
$S_{0R}(3,6)$	-	48	84	136	36	32	24	12	-	6	-	-	-	-	-	-	-
$S_{1B}(3,5)$	48	104	100	18	10	-	-	-	-	-	-	-	2	-	-	-	-

	4	5	6	7	8	9	10
$s_{0B}(3,5)$	-	-	8	18	4	2	-
$s_{1R}(3,5)$	-	182	48	20	4	-	2
$s_{0R}(3,5)$	-	48	44	42	2	2	-
$s_{1B}(4,5)$	-	144	120	-	2	-	-
$s_{0B}(4,5)$	-	32	64	-	2	-	-
$s_{1R}(4,5)$	-	336	84	4	-	-	-
$s_{0R}(4,5)$	-	208	108	4	-	-	-
$s_{1B}(3,4)$	20	8	8	-	-	-	-

	3	4	5
$s_{0B}(3,4)$	-	-	4
$s_{1R}(3,4)$	-	40	14
$s_{0R}(3,4)$	-	24	14
$s_{1B}(4,4)$	-	8	-
$s_{0B}(4,4)$	-	8	-
$s_{1R}(4,4)$	-	24	-
$s_{0R}(4,4)$	-	24	-
$s_{1R}(3,3)$	8	-	-

	2	3	4	5	6
$s_{0R}(3,3)$	-	8	-	-	-
$s_{1B}(2,4)$	-	-	-	4	8
$s_{1R}(2,4)$	-	-	8	-	-
$s_{0R}(2,4)$	-	-	-	-	2
$s_{1R}(2,3)$	-	8	-	-	-
$s_{0R}(2,3)$	-	-	2	-	-
$s_{1R}(2,2)$	2	-	-	-	-
$s_{0R}(2,2)$	2	-	-	-	-

Gesamt:

	0	1								
0	-	2	4	24	158	1052	5350	19916	48874	71502
1	81436	67378	54370	39034	29068	21066	15070	10192	8170	5940
2	4378	3396	2722	1984	1586	1286	964	706	592	490
3	414	312	290	254	222	170	112	78	98	102
4	74	48	58	52	42	32	30	24	24	26
5	16	20	26	12	6	10	14	12	4	6
6	6	6	8	6	10	4	2	4	10	8
7	2	2	-	-	2	2	-	-	4	-
8	-	-	-	2	2	-	-	-	-	-
9	-	-	-	-	-	-	4	2	-	-
10	-	-	-	-	-	-	-	2	-	-

Statistik über Anzahl Einsen auf dem Band nach dem Halt

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$s_{1B}(4,8)$	7566	31524	58079	62092	42557	19816	6722	1854	436	85	16	8	5	2
$s_{0B}(4,8)$	2530	10260	16886	14884	7826	2702	756	189	41	6	2	-	-	-
$s_{1R}(4,8)$	3497	13472	20587	15508	5676	849	72	3	-	-	-	-	-	-
$s_{0R}(4,8)$	3989	13058	16054	9275	2664	418	45	1	-	-	-	-	-	-
$s_{1B}(4,7)$	2145	8265	13694	12353	6485	2106	441	49	9	1	-	-	-	-
$s_{0B}(4,7)$	689	2653	3786	2567	936	211	21	1	-	-	-	-	-	-
$s_{1R}(4,7)$	1622	5620	7362	4495	1239	108	-	-	-	-	-	-	-	-
$s_{0R}(4,7)$	1609	4766	5024	2256	438	52	3	-	-	-	-	-	-	-
$s_{1B}(4,6)$	359	1160	1561	1107	422	85	10	-	-	-	-	-	-	-
$s_{0B}(4,6)$	136	386	426	215	43	4	-	-	-	-	-	-	-	-
$s_{1R}(4,6)$	345	1009	1198	667	138	5	-	-	-	-	-	-	-	-
$s_{0R}(4,6)$	379	928	778	257	31	3	-	-	-	-	-	-	-	-
$s_{1B}(3,6)$	63	274	470	413	208	59	5	-	-	-	-	-	-	-
$s_{0B}(3,6)$	13	45	62	43	15	2	-	-	-	-	-	-	-	-
$s_{1R}(3,6)$	57	184	223	111	15	-	-	-	-	-	-	-	-	-

	0	1	2	3	4	5
$S_{0R}(3,6)$	48	142	135	47	6	-
$S_{1B}(4,5)$	26	77	93	54	14	2
$S_{0B}(4,5)$	17	37	31	12	1	-
$S_{1R}(4,5)$	48	149	153	63	11	-
$S_{0R}(4,5)$	67	140	92	20	1	-
$S_{1B}(3,5)$	19	65	100	73	22	3
$S_{0B}(3,5)$	4	12	11	4	1	-
$S_{1R}(3,5)$	31	100	101	33	1	-
$S_{0R}(3,5)$	25	59	43	10	1	-
$S_{1B}(4,4)$	1	3	3	1	-	-
$S_{0B}(4,4)$	1	3	3	1	-	-
$S_{1R}(4,4)$	2	7	9	5	1	-
$S_{0R}(4,4)$	5	11	7	1	-	-

	0	1	2	3	4	5
$S_{1B}(3,4)$	4	12	13	6	1	-
$S_{0B}(3,4)$	1	2	1	-	-	-
$S_{1R}(3,4)$	10	21	17	6	-	-
$S_{0R}(3,4)$	11	19	8	-	-	-
$S_{1B}(2,4)$	-	2	5	4	1	-
$S_{1R}(2,4)$	2	4	2	-	-	-
$S_{0R}(2,4)$	-	1	1	-	-	-
$S_{1R}(3,3)$	1	3	3	1	-	-
$S_{0R}(3,3)$	2	4	2	-	-	-
$S_{1R}(2,3)$	2	4	2	-	-	-
$S_{0R}(2,3)$	1	1	-	-	-	-
$S_{1R}(2,2)$	-	1	1	-	-	-
$S_{0R}(2,2)$	1	1	-	-	-	-

Gesamt:

0	1	2	3	4	5	6	7	8	9	10	11	12	13
25329	94485	147026	126584	68754	26425	8075	2097	486	92	18	8	5	2

Statistik über Feldweite (Anzahl der benötigten Arbeitsfelder)

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$S_{1B}(4,8)$	-	21240	76272	74516	40850	12882	3714	904	240	92	34	12	4	-	2
$S_{0B}(4,8)$	-	8490	21540	16106	6710	2264	744	152	40	28	8	-	-	-	-
$S_{1R}(4,8)$	2068	23268	26808	6586	800	124	8	-	2	-	-	-	-	-	-
$S_{0R}(4,8)$	1312	14134	21140	7440	1238	220	14	4	-	2	-	-	-	-	-
$S_{1B}(4,7)$	-	8744	19560	12210	4442	524	58	6	4	-	-	-	-	-	-
$S_{0B}(4,7)$	-	3230	4964	2160	442	62	6	-	-	-	-	-	-	-	-
$S_{1R}(4,7)$	1646	10550	7062	1140	44	4	-	-	-	-	-	-	-	-	-
$S_{0R}(4,7)$	988	6082	5640	1372	62	4	-	-	-	-	-	-	-	-	-
$S_{1B}(4,6)$	-	1734	2098	722	150	-	-	-	-	-	-	-	-	-	-
$S_{0B}(4,6)$	-	702	418	84	6	-	-	-	-	-	-	-	-	-	-
$S_{1R}(4,6)$	550	1782	976	54	-	-	-	-	-	-	-	-	-	-	-
$S_{0R}(4,6)$	342	1182	780	72	-	-	-	-	-	-	-	-	-	-	-
$S_{1B}(3,6)$	-	436	686	330	36	4	-	-	-	-	-	-	-	-	-
$S_{0B}(3,6)$	-	46	102	26	6	-	-	-	-	-	-	-	-	-	-
$S_{1R}(3,6)$	118	390	80	2	-	-	-	-	-	-	-	-	-	-	-

	2	3	4	5
$s_{0R}(3,6)$	50	214	110	4
$s_{1B}(4,5)$	-	130	104	32
$s_{0B}(4,5)$	-	82	16	-
$s_{1R}(4,5)$	100	252	72	-
$s_{0R}(4,5)$	76	168	76	-
$s_{1B}(3,5)$	-	154	102	26
$s_{0B}(3,5)$	-	18	14	-
$s_{1R}(3,5)$	90	166	10	-
$s_{0R}(3,5)$	40	86	12	-

	2	3	4
$s_{1B}(4,4)$	-	8	-
$s_{0B}(4,4)$	-	8	-
$s_{1R}(4,4)$	8	8	8
$s_{0R}(4,4)$	8	8	8
$s_{1B}(3,4)$	-	28	8
$s_{0B}(3,4)$	-	4	-
$s_{1R}(3,4)$	30	24	-
$s_{0R}(3,4)$	-	18	20
$s_{1B}(2,4)$	-	8	4

	2	3	4
$s_{1R}(2,4)$	-	8	-
$s_{0R}(2,4)$	-	2	-
$s_{1R}(3,3)$	4	4	-
$s_{0R}(3,3)$	4	4	-
$s_{1R}(2,3)$	8	-	-
$s_{1R}(2,3)$	2	-	-
$s_{1R}(2,2)$	2	-	-
$s_{0R}(2,2)$	2	-	-

Gesamt:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	7448	103412	188690	122882	54786	16088	4544	1066	286	122	42	12	4	-	2

In Tabelle 3 sind alle Turingmaschinen aus $\mathcal{R}_4$ zusammengefaßt, die nur die rechte Seite des Bandes benutzen. Die Belegung des Bandes nach dem Halt wird als Dualzahl aufgefaßt: Feld 0 entspricht 2^0, Feld 1 entspricht 2^1 usw. Es gibt also zum Beispiel 81 verschiedene Turingmaschinen aus $\mathcal{R}_4$, die nur die rechte Seite des Bandes benutzen und die in obigem Sinne die Zahl 63 auf dem Band hinterlassen, d.h. eine Belegung folgender Art:

Feld	0	1	2	3	4	5	6	...	
Belegung	1	1	1	1	1	1	0	...	
entsprechend	2^0	$+2^1$	$+2^2$	$+2^3$	$+2^4$	$+2^5$			$= 63$

	0	1	2	3	4	5	6	7	8	9
0	11755	11018	12431	13088	10687	10315	12014	13268	4667	3554
1	4565	4620	5290	4617	5743	6657	822	452	596	454
2	696	513	662	686	916	598	817	661	856	682
3	931	1081	69	13	42	28	50	26	45	46
4	76	36	67	38	57	32	50	49	96	34
5	70	38	73	52	77	68	107	40	65	47
6	83	49	66	81	9	2	2	2	5	2
7	4	1	9	3	4	4	4	-	4	4
8	7	-	4	3	4	2	5	3	1	1
9	5	3	3	-	5	8	11	4	3	-
10	4	2	9	2	13	1	10	3	2	3
11	8	3	10	2	6	4	3	2	14	2
12	13	1	13	7	2	2	14	3	2	-

Außerdem tauchen folgende Zahlen auf:

135	170	216	246	488
139	171	224	252	949
144	182	228	426	1290
151	192(2x)	236	437	1354
152	201	244	456	

Tabelle 3

Im weiteren wird noch eine Kollektion von "maximalen" Turingmaschinen zusammengestellt. Zunächst werden aus $S_{1B}(3,6)$ alle Turingmaschinen aufgeführt, die eine Laufzeit größer als 17 oder eine Feldweite gleich 7 besitzen oder 6 Einsen auf dem Band hinterlassen. Anschließend werden aus $S_{10}(4,8)$ alle Turingmaschinen aufgeführt mit einer Laufzeit größer als 73 oder eine Feldweite größer als 12 oder die mehr als 10 Einsen auf dem Band hinterlassen. Bei den Turingmaschinen aus $S_{10}(5,10)$, die dann folgen, handelt es sich um solche, die eine Laufzeit größer als 900 besitzen und mehr als 34 Einsen hinterlassen und die in "intuitiven Stichproben" von einigen Millionen Turingmaschinen vorkamen. Die letzte Turingmaschine, die 40 Einsen erzeugt, übertrifft nicht nur die bisherige beste Abschätzung für n=5 ($\Sigma(5) \geq 17$; siehe Kapitel I) um mehr als das doppelte, sondern sogar noch die Abschätzung für n=6 ($\Sigma(6) \geq 35$; siehe Einleitung).

Einige Erklärungen zu den folgenden Aufstellungen: Unter dem Funktionswert von φ befinden sich in Klammern die Nummern derjenigen Felder einschließlich, zwischen denen alle Arbeitsfelder beim Ablauf gelegen haben. Also (-6/4) heißt: Als Arbeitsfelder wurden die Felder -6 bis 4 einschließlich benutzt. d_1 ist diejenige Zahl (dezimal), die entsteht, wenn man die Belegung der linken Seite des Bandes (ab Feld -1) als Dualzahl auffaßt. Dasselbe gilt analog für d_2 und die rechte Seite des Bandes. Die Zahl in Klammern gilt für diejenige Turingmaschine, die aus der angegebenen dadurch entsteht, daß sie im Haltebefehl eine 0 schreibt; damit läßt sich auch ablesen, bei welchem Arbeitsfeld die Turingmaschine anhält. Steht zum Beispiel unter d_1/d_2: 22(6)/13, sieht die Belegung folgendermaßen aus:

	d_1						d_2					
Feld ...	-6	-5	-4	-3	-2	-1	0	1	2	3	4	...
Belegung ...	0	1	0	1	1	0	1	0	1	1	0	...
entsprechend ...		2^4	+	2^2+	2^1		2^0	+	2^2+2^3			

Da für d_1 in Klammern 6(=22-16) angegeben ist, hält die Turingmaschine mit dem Arbeitsfeld -5 an.

Befehl 1	Befehl 2	Befehl 3	Befehl 4	σ	φ	β	d_1/d_2
101R2 111L3	201R3 211R0	301L1 310L2		11	6 (-3/2)	6	7(5)/7
101R2 111R1	201L3 211R0	301R1 311L2		12	6 (-2/3)	6	3/15(13)
101R2 110R3	201L3 211R0	301R1 310L2		13	7 (-2/4)	4	2/21(17)
101R2 111R3	201L3 211R0	301R1 310L2		13	7 (-2/4)	6	3/23(19)
101R2 111R0	200R3 211R2	301L3 311L1		14	6 (-1/4)	6	1/31(29)
101R2 111L3	201L1 211R2	301L2 311R0		13	6 (-3/2)	6	7/7(6)
101R2 110R1	200R3 211R0	301L3 310L1		19	5 (-1/3)	4	1/14(10)
101R2 111R1	200R3 211R0	301L3 310L1		19	5 (-1/3)	5	1/15(11)
101R2 111R0	200L3 210R3	301L3 311L1		20	5 (-2/2)	5	3(2)/7
101R2 111L1	200R3 211R0	301L3 310L1		20	6 (-2/3)	5	3(2)/13
101R2 111R0	201L2 210R3	301L3 311L1		21	5 (-1/3)	5	1/15(14)

Befehl				σ	φ	β	d_1/d_2
1	2	3	4				
101R2	200R3	301R4	401L3	31	13	7	2/1358(1294)
110L4	210R2	311R1	411R0		(-2/10)		
101R2	200R3	301R4	401L1	40	13	10	55(51)/121
111L4	211R2	311R0	410L4		(-6/6)		
101R2	201L1	300L2	401R1	40	11	11	63(62)/31
110L3	211R4	311R2	411R0		(-6/4)		
101R2	201R3	301L1	401L3	42	13	7	690(674)/6
110L3	210R1	310L4	411R0		(-10/2)		
101R2	201R3	301L4	401L2	43	11	11	15/127(123)
110L1	211R0	310R1	411L4		(-4/6)		
101R2	201L3	301R1	401R1	46	11	11	7/255(253)
110R3	211R0	311L4	411L2		(-3/7)		
101R2	201L3	301R1	401R0	53	12	12	7/511(255)
111L2	211R4	311L1	410R3		(-3/8)		
101R2	201R3	301L1	401R3	59	12	11	55(53)/63
111L4	211R0	310R1	410L4		(-6/5)		
101R2	201L3	300L4	401R1	61	16	9	26/1365(1349)
110R1	210R4	310L2	411R0		(-5/10)		
101R2	201R3	301L2	401R0	63	12	12	7/511(255)
110L1	211L2	311R4	410R1		(-3/8)		
101R2	201L3	301R0	401R1	74	9	8	7/47(15)
110R3	211R1	310L4	411L1		(-3/5)		
101R2	200R3	301L3	400R2	75	13	9	3/1987(1923)
110R4	210R1	310L1	411R0		(-2/10)		
101R2	200R3	301L3	401R0	78	10	9	23(19)/31
111L1	210R4	310L1	410R3		(-5/4)		
101R2	201L3	301R1	401R0	78	13	12	31/254(252)
110R4	210L1	311L2	410R3		(-5/7)		
101R2	201L3	301L1	400L3	83	9	8	63(31)/3
111R0	210R4	311L2	411R4		(-6/2)		
101R2	200L3	301L3	401R0	84	13	11	446(190)/15
111L4	210R3	311L1	410L1		(-9/3)		

Befehl				σ	φ	β	d_1/d_2
1	2	3	4				
101R2 110R3	201L1 211R1	301RO 311R4	401L4 410L2	96	14 (-2/11)	13	3/3071(1023)
101R2 111L2	200L3 210R4	301RO 311L1	401R1 410L1	96	10 (-5/4)	6	22(6)/13
101R2 111L4	201L3 210R2	301R1 311L1	401RO 410L3	97	12 (-4/7)	9	14(6)/187
101R2 111L2	201L1 210L3	301RO 311L4	401R4 410R1	107	14 (-10/3)	13	767(255)/15

Befehl					σ	φ	β
1	2	3	4	5			
101R2 111L4	201R3 210R5	301L1 311RO	401L3 410L4	500R2 511L1	919	54 (-42/11)	34
101R2 110R3	201R3 210R5	301L4 311R3	401L1 410L4	501R1 511RO	978	23 (-5/17)	17
101R2 110L4	201R3 210R2	301L1 310R1	400L3 410L5	501L3 411RO	992	45 (-31/13)	23
101R2 111L1	201R3 210R5	301R4 310R4	401L1 411R4	501RO 510L1	532	36 (-4/31)	35
101R2 110L5	201R3 210R1	301L4 311RO	400L5 411L4	501L1 510L2	530	47 (-35/11)	35
101R2 111L1	201R3 210R5	301L4 310R4	401L1 411R4	501RO 510L1	540	36 (-4/31)	35
101R2 110L5	201R3 210R1	301L4 311RO	400L5 410R5	501L1 511L3	544	47 (-35/11)	35
101R2 110R5	201R3 211L5	301R4 310R1	401L2 411RO	501L2 510L5	739	50 (-36/13)	35
101R2 111L1	201R3 211R5	301R4 310R4	401L1 411R4	501RO 511R3	484	36 (-4/31)	36
101R2 111L1	201R3 211R5	301L4 310R4	401L1 411R4	501RO 511R3	492	36 (-4/31)	36

		Befehl			σ	φ	β
1	2	3	4	5			
101^{R2}	201R3	301R4	401L2	501L1	771	42	36
111^{L1}	210L5	310R1	411R0	511L2		(-15/26)	
101^{R2}	201R3	301R4	400R5	501L1	803	43	36
111^{L1}	210L5	310R1	411R0	511L2		(-15/27)	
101^{R2}	201R3	301L4	401R5	500L2	714	38	37
111^{R0}	210L3	311R1	410L4	510R3		(-24/13)	
101^{R2}	201R3	301L4	401L2	501R0	939	55	37
111^{R5}	211R2	310R1	411L4	511R3		(-8/46)	
101^{R2}	201R3	301L4	401L4	501R0	955	37	37
111^{L1}	210R3	311L1	411L5	510L1		(-30/6)	
101^{R2}	201R3	301R4	400L5	501L3	974	53	37
110^{R5}	211R0	310L3	410R1	511L5		(-37/15)	
101^{R2}	201R3	301L4	401L5	500R1	556	44	40
111^{R1}	210R2	311L3	410L3	511R0		(-15/28)	

VI Anmerkungen zur praktischen Durchführung

Für die praktische Bestimmung von $\mathcal{R}_4$ wurden folgende Teilmengen von $\mathcal{R}_4$ gebildet:

$$S^i = \bigcup_{\nu=[i/2+1]}^{4} S(\nu,i) \qquad U^i = \bigcup_{\nu=[i/2+1]}^{4} U(\nu,i)$$

In S^i wurden alle Turingmaschinen aus $\mathcal{R}_4$ zusammengefaßt, die halten und genau i Teilbefehle während des Ablaufs benutzen. U^i besteht aus allen nicht-haltenden Turingmaschinen aus $\mathcal{R}_4$, die während des Ablaufs genau i Teilbefehle benutzen. Ein Computerprogramm, dem $S^1=S(1,1)$ (vgl. Seite 15) vorgegeben wurde, bestimmte sukzessiv $\hat{S}^2,\hat{U}^2,\hat{S}^3,\hat{U}^3,\ldots,\hat{S}^8$, wobei anschließend nachgewiesen wurde, daß gilt: $\hat{S}^i=S^i$ und $\hat{U}^i=U^i$. In den Mengen $\hat{S}^i$ wurden immer die beiden Turingmaschinen, die sich nur in dem einzigen expliziten Haltebefehl unterscheiden (der Haltebefehl kann 0 oder 1 schreiben), zu einem Paar zusammengefaßt. Ist $\hat{S}^i$ bestimmt, so ergeben sich $\hat{S}^{i+1}$ und $\hat{U}^i$ folgendermaßen: Das Computerprogramm liest von jedem Paar in $\hat{S}^i$ nacheinander jeweils eine Turingmaschine ein, simuliert den Ablauf der eingelesenen Turingmaschine bis zum Halt und variiert den Haltebefehl wie im Konstruktionsverfahren in Kapitel III angegeben. Die Unterscheidung in Fall 1 und Fall 2 braucht nicht getroffen zu werden, da S(n-1,m-1) und S(n,m-1) beide Teilmengen von $\hat{S}^{m-1}$ sind. Es muß nur darauf geachtet werden, daß als nächster durchzuführender Befehl nicht ein solcher genommen wird, der eine höhere Nummer hat als ein Befehl der Form $\begin{matrix} i0* \\ i1* \end{matrix}$. Ist z.B. $\begin{matrix} 30* \\ 31* \end{matrix}$ vorhanden, darf die Nummer des als nächsten durchzuführenden Befehls höchstens 3 sein. Die Zwischenbelegung des Bandes und die Stellung des Lese- und Schreibkopfes beim Erreichen des Haltebefehls wurden abgespeichert, so daß der weitere Ablauf nach Variation des Haltebefehls jeder der entstehenden Turingmaschinen simuliert werden

konnte. Die Entscheidung, ob eine Turingmaschine zur Menge $\hat{S}^{i+1}$ oder $\hat{U}^{i}$ kam, wurde folgendermaßen getroffen: Überstieg die Gesamtschrittzahl (Schrittzahl bis zum Halt plus Anzahl der Schritte nach Variation des Haltebefehls) eine von i abhängige vorgegebene Schranke (z.B. 200 bei i=7), ohne daß ein Befehl der Form iZ* erreicht wurde, kam diese Turingmaschine zur Menge $\hat{U}^{i}$. Wurde ein Befehl der Form iZ* erreicht, wurde dieser geändert in iZ0R0 bzw. iZ1R0 und diese beiden Turingmaschinen kamen als Paar zur Menge $\hat{S}^{i+1}$.

Bei der Simulation des Ablaufs diente ein einfach dimensioniertes Zahlenfeld als Arbeitsband und ein Index markierte die Stellung des Lese- und Schreibkopfes auf dem Band. Ein Speicherplatz des Zahlenfeldes entsprach einem Feld des Bandes und konnte mit 0 oder 1 belegt werden. Das Turingprogramm wurde in zwei zwei-dimensionalen Zahlenfeldern abgespeichert. Im ersten wurden die Schreib- und Verschiebeoperationen abgespeichert und im zweiten die Nummern der als nächste durchzuführenden Befehle. Zu jedem Zahlenfeld gab der erste Index die Nummer des Befehls an. Der zweite Index konnte die Werte 0 oder 1 haben, wobei eine 0 dem ersten Teilbefehl eines Befehls und eine 1 dem zweiten entsprach. Die Schreib- und Verschiebeoperationen eines jeden Teilbefehls wurden zusammen durch eine Zahl ausgedrückt: 1 für 0L, 2 für 0R, 3 für 1L, 4 für 1R und 0 für einen Teilbefehl der Form iZ*. Im zweiten Zahlenfeld waren die Nummern der Befehle abgespeichert, die von dem Teilbefehl aufgerufen wurden, der zu der entsprechenden Kombination von erstem und zweitem Index gehörte. Die Abspeicherung eines Turingprogramms veranschaulicht Abbildung 3a. Bei der Simulation des Ablaufs einer Turingmaschine wird der erste Index durch das Zahlenfeld B verändert und der zweite durch die Belegung des markierten Speicherplatzes des Zahlenfeldes, das das Arbeitsband darstellt.

101R2
111L4
201L3
201R2
301R1
311L1
401R0
410L3

Zahlenfeld

A

	1	2
1	4	3
2	3	4
3	4	3
4	4	1

B

	1	2
1	2	4
2	3	2
3	1	1
4	0	3

Abb. 3a

Die Teilmengen $\hat{S}^1$ bis $\hat{S}^8$ und $\hat{U}^1$ bis $\hat{U}^7$ wurden als Dateien auf Magnetband abgespeichert. Zu jedem Turingprogramm aus einer der Mengen $\hat{S}^i$ wurden die Funktionswerte σ, φ und β und die Bandbelegung beim Halt durch zwei Dualzahlen wie in Kapitel V dargestellt - mit abgespeichert. Für die Aufstellung der Statistiken in Kapitel V lasen Computerprogramme die acht Dateien entsprechend den Teilmengen $\hat{S}^1$ bis $\hat{S}^8$ ein, entschieden von jeder Turingmaschine, in welche Teilmenge S(n,m) sie gehörte, was leicht durch die Belegung der einzelnen Teilbefehle durch explizite Teilbefehle bzw. durch Teilbefehle der Form iZ* zu ermitteln war, und teilten die Turingmaschinen nach den abgespeicherten Funktionswerten in die entsprechenden Kategorien ein.

Um zu beweisen, daß alle Turingmaschinen aus $\hat{U}^i$ nicht halten, wurden zu den 4 Nichthaltekriterien 4 Computerprogramme geschrieben. Bei Anwendung eines Computerprogramms, entsprechend der Anwendung eines Nichthaltekriteriums, wurden aus der Menge $\hat{U}^i$ die Turingmaschinen entfernt, die das zugehörige Nichthaltekriterium erfüllten. In Tabelle 4 ist angegeben, wie sich durch Anwendung der 4 Nichthaltekriterien die Anzahl der Turingmaschinen verringerte, von der das Nichthalten noch nicht nachgewiesen ist. Die Turingmaschinen aus U^8 halten trivialerweise nicht, da sie keinen Halte-

befehl besitzen. Deshalb ist U^8 in der Tabelle nicht mit aufgeführt. Die Indizierungen OR, 1R, OB, 1B haben dieselbe Bedeutung wie in Definition 7.

U^i_{OR}

i	Anzahl	nach Anwendung von Kriterium 1	2	3	4
2	5	-	-	-	-
3	24	-	-	-	-
4	189	2	-	-	-
5	1409	11	7	5	-
6	9922	164	62	47	-
7	60800	1278	1007	466	40
	72340	1455	1076	518	40

U^i_{1R}

i	Anzahl	nach Anwendung von Kriterium 1	2	3	4
2	4	-	-	-	-
3	27	-	-	-	-
4	229	-	-	-	-
5	1890	8	-	-	-
6	13797	96	50	33	-
7	84653	809	372	181	8
	100600	913	422	214	8

U^i_{OB}

i	Anzahl	nach Anwendung von Kriterium 1	2	3	4
2	2	-	-	-	-
3	14	-	-	-	-
4	121	-	-	-	-
5	1223	16	8	5	-
6	10724	373	179	60	-
7	88504	4783	3361	1958	92
	100588	5172	3548	2023	92

U^i_{1B}

i	Anzahl	nach Anwendung von Kriterium 1	2	3	4
2	-	-	-	-	-
3	14	-	-	-	-
4	248	4	-	-	-
5	2940	51	40	27	-
6	34390	1033	391	186	2
7	298086	12044	9822	5334	254
	335678	13132	10253	5547	256

Tab. 4

Zu Kriterium 1: Zu jeder Turingmaschine wurden eine bestimmte Anzahl Schritte simuliert (z.B. 100 bei $\hat{U}^7$), die Bandbelegung und die Nummer des zuletzt durchgeführten Befehls abgespeichert und bei den nächsten Schritten (z.B. 50 weitere Schritte bei $\hat{U}^7$), die simuliert wurden, untersucht, ob der im Augenblick durchgeführte Befehl mit dem abgespeicherten identisch ist und wenn ja, ob die Bandbelegung soweit mit der abgespeicherten Bandbelegung übereinstimmt, daß Kriterium 1 erfüllt ist. Die Werte l_{nm}, r_{nm} des Kriteriums 1 wurden bei jedem simulierten Schritt mitgeführt und entsprechend verändert.

Zu Kriterium 2: Die Umsetzung dieses Kriteriums in ein Computerprogramm ist trivial.

Zu Kriterium 3: An Hand der Abb. 3 kann die Umsetzung dieses Kriteriums in ein Computerprogramm nachvollzogen werden.

Zu Kriterium 4: Hierzu ein Computerprogramm zu entwickeln, war die schwerste Aufgabe, da der in der Formulierung des Kriteriums benutzte Begriff des Wendepunktes sich nicht eindeutig aus dem Ablauf einer Turingmaschine ergibt. Das Computerprogramm ging folgendermaßen vor: Es simulierte den Ablauf einer Turingmaschine so lange, bis sich der Lese- und Schreibkopf zum ersten Mal 8 Felder weit (nach links oder rechts) vom Startfeld (Feld 0) wegbewegt hatte. Wurde etwa das Feld 8 eher als das Feld -8 erreicht, wurde angenommen, daß die sich vergrößernde Schleife zumindest auch nach rechts wandert. Als erster rechter Wendepunkt wurde dann das Arbeitsfeld genommen, bei dem der Lese- und Schreibkopf, nachdem er das Feld 8 erreicht hat, zum ersten Mal nach links verschoben wurde. Liegt so ein erster rechter Wendepunkt fest, wird ein zweiter rechter Wendepunkt dadurch definiert, daß der Lese- und Schreibkopf, nachdem er den ersten rechten Wendepunkt nach rechts überschritten hat, zum ersten Mal wieder nach links verschoben wird. Zwischen diesen bei-

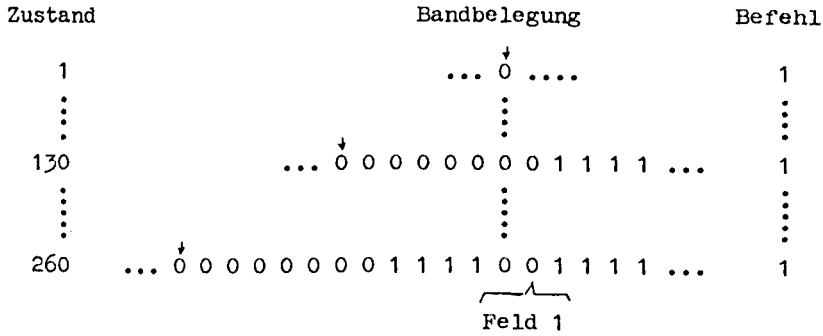
den Wendepunkten ergibt sich dann als linker Wendepunkt das am weitesten links gelegene Arbeitsfeld zwischen den beiden rechten Wendepunkten. Nach diesem Verfahren wurde dann eine Folge von rechten und linken (bzw. im Falle, daß das Feld -8 eher erreicht wurde als das Feld 8, eine Folge von linken und rechten) Wendepunkten ermittelt. Diese wurden dann daraufhin untersucht, ob der Abstand von je zwei aufeinanderfolgenden rechten Wendepunkten konstant war; dasselbe wurde für linke Wendepunkte untersucht. War dies nicht der Fall, wurde versucht, durch Streichung jedes zweiten Paares von linkem und rechtem Wendepunkt diesen Zustand zu erreichen. Es wurde dann geprüft, ob diese Folge von Wendepunkten mit den dazu abgespeicherten Informationen über Bandbelegung und Nummer des beim Wendepunkt durchgeführten Befehls das Kriterium 4 erfüllte.

Bei insgesamt 396 Turingmaschinen (vgl. Tab. 4) aus $\hat{U}^7$ und $\hat{U}^6$ konnte durch keines der 4 Computerprogramme das Nichthalten bewiesen werden. Von jeder dieser Turingmaschinen wurden die ersten 300 Schritte simuliert und zu jedem Schritt Bandbelegung und Nummer des durchgeführten Befehls auf dem Zeilendrucker ausgegeben. Für jede dieser Turingmaschinen konnte dann an Hand der Liste der ersten 300 Schritte das Nichthalten "per Hand" festgestellt werden. Am Beispiel einiger charakteristischer Fälle soll dieses "per Hand" verdeutlicht werden.

Eine größere Zahl der 396 Turingmaschinen erfüllte das Kriterium 1, wurde aber vom zugehörigen Programm nicht "entdeckt", da die Laufzeit bis zum Eintreten der Schleife und die Schleifengröße derart groß waren, daß sie die Grenzen des Programms sprengten. Als Beispiel dazu diene folgende Turingmaschine:

1	2	3	4
101R2	200R3	301L3	400R1
110L1	211R0	311L4	411L1

Die Schleife besteht bei dieser Turingmaschine aus 130 Schritten (S(4) = 107!)



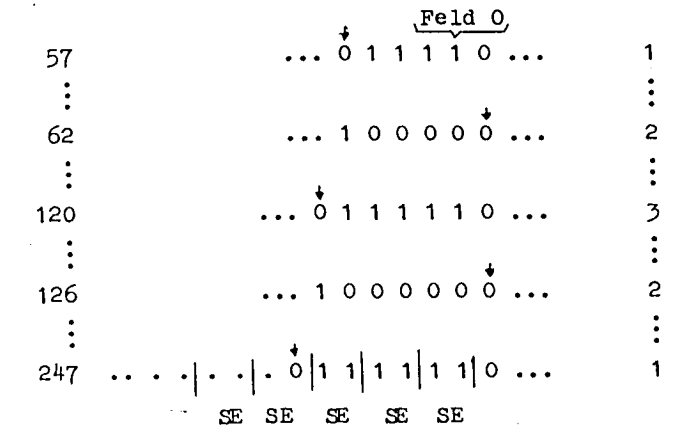
Das zwischen Zustand 130 und 260 am weitesten rechts gelegene Arbeitsfeld ist Feld 1. Die Punkte bei der Bandbelegeung bedeuten, daß diese Felder noch nicht als Arbeitsfelder benutzt wurden.

Bei den Turingmaschinen, die keines der vier Kriterien erfüllen, konnte das Nichthalten durch Modeifizierung des Kriteriums 4 gezeigt werden. Dies mögen zwei charakteristische Beispiele veranschaulichen.

Im ersten Beispiel vergrößert sich die Schleife während zweier aufeinanderfolgender linker Wendepunkte um ein Feld. Trotzdem besteht hier ein Symmetrieelement aus zwei Feldern. Außerdem nimmt jedes Symmetrieelement beim Ablauf vom rechten zum linken Wendepunkt vier Zustände und Belegungen an (bei Kriterium 4 nur 3 Zustände).

1	2	3	4
101R2	200L1	301R4	401R4
111L3	210R4	311L1	410R2

Zunächst einige aufeinanderfolgende Wendepunkte:



Man teile vom Feld 0 an nach links das gesamte Band, unabhängig davon, ob die Felder schon als Arbeitsfelder benutzt wurden, in Symmetrieelemente von je zwei Feldern ein. Die oben erwähnten 4 Zustände eines jeden solchen Symmetrieelementes sind:

1) Anfangszustand mit der Belegung	...|0 0|...	
2) Der Lese- und Schreibkopf erreicht das Symmetrieelement zum 1. Mal von rechts und verläßt es wieder nach rechts:	...|0 0|...	1
	...|0 1|...	2
3) Der Lese- und Schreibkopf erreicht das Symmetrieelement zum 2. Mal von rechts und verläßt es wieder nach rechts:	...|0 1|...	1
	...|0 1|...	3
	...|1 1|...	4
	...|1 0|...	2
4) Der Lese- und Schreibkopf erreicht das Symmetrieelement zum 3. Mal von rechts und verläßt es nach links	...|1 0|...	1
	...|1 1|...	3
	...|1 1|...	1

Erreicht der Lese- und Schreibkopf das Symmetrieelement von links, verläßt er es nach rechts und der Ausgangszustand wird wieder hergestellt:

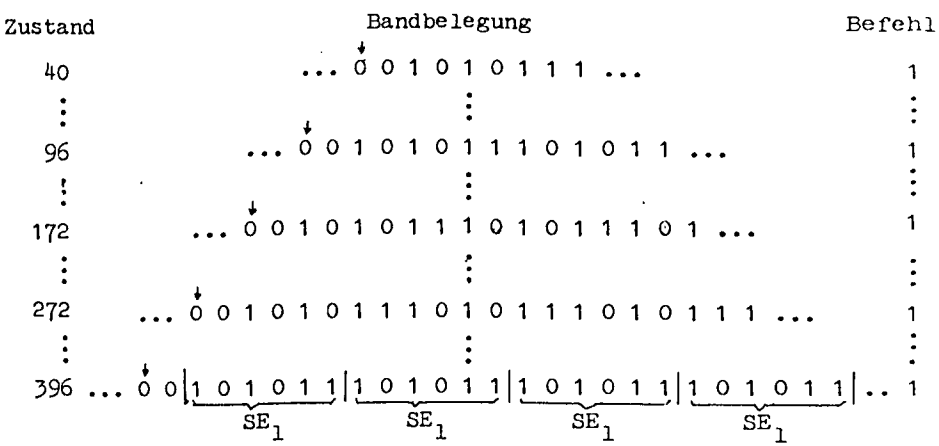
```
...|1 1|...      2
...|0 1|...      4
...|0 0|...      2
```

Jedes Symmetrieelement wird immer mit Befehl 1 von links erreicht und immer mit Befehl 2 von rechts. Da sich daran auch bei Erreichen des Feldes 1 nichts ändert, hält diese Turingmaschine nicht an.

Das letzte Beispiel gehört zu den komplizierteren Fällen. Das Symmetrieelement besteht aus 6 Feldern. Zwischen zwei aufeinanderfolgenden linken Wendepunkten vergrößert sich die Schleife um 5, 3 oder 4 Felder. Nach jeweils 3 linken Wendepunkten, entsprechend einer Schleifenvergrößerung von 12 Feldern, sind genau zwei Symmetrieelemente dazugekommen.

1	2	3	4
101R2	200R3	301L4	400L1
110L2	210R4	311R1	411R0

Zunächst einige aufeinanderfolgende linke Wendepunkte:



Beim Ablauf vom linken zum rechten Wendepunkt wird jedes Symmetrieelement SE_l einmal von links erreicht und nach rechts verlassen, und zwar immer mit dem Befehl 3:

$$\begin{array}{ll} \ldots|\underset{\downarrow}{1}\ 0\ 1\ 0\ 1\ 1|\ldots & 3 \\ \ldots|1\ \underset{\downarrow}{0}\ 1\ 0\ 1\ 1|\ldots & 1 \\ \vdots & \\ \ldots|1\ 1\ 1\ 0\ 1\ \underset{\downarrow}{0}|\ldots & 2 \\ \ldots|1\ 1\ 1\ 0\ 1\ 0|\underset{\downarrow}{\ldots} & 3 \end{array}$$

Beim Ablauf vom rechten zum linken Wendepunkt wird folgender Abschnitt als Symmetrieelement SE_r genommen:

$$\ldots|0\ 1\ 1\ 1\ 0\ 1|\ldots$$

Wird SE_r mit dem Befehl 1 von rechts her erreicht, wird es ebenfalls mit Befehl 1 nach links verlassen und die Belegung ist dann:

$$\ldots|0\ 1\ 0\ 1\ 1\ 1|\ldots$$

entspricht also wieder der Belegung für den Zustand am linken Wendepunkt. Wird rechts das erste noch nicht als Arbeitsfeld benutzte Feld erreicht, so müssen drei Fälle unterschieden werden:

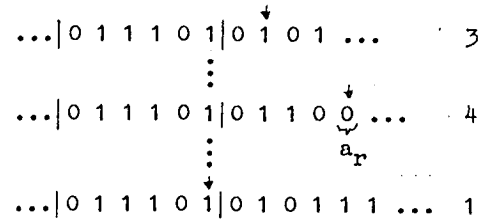
1)

$$\begin{array}{ll} & \overset{SE_r}{\overbrace{}} \\ \ldots|0\ 1\ 1\ 1\ 0\ 1|0\ \underset{\downarrow}{\underbrace{0}}\ \ldots & 3 \\ & a_r \\ \ldots|0\ 1\ 1\ 1\ 0\ 1|\underset{\downarrow}{0}\ 1\ \ldots & 4 \\ \ldots|0\ 1\ 1\ 1\ 0\ \underset{\downarrow}{1}|0\ 1\ \ldots & 1 \end{array}$$

2)

$$\begin{array}{ll} \ldots|0\ 1\ 1\ 1\ 0\ 1|0\ \underset{\downarrow}{1}\ \ldots & 3 \\ \ldots|0\ 1\ 1\ 1\ 0\ 1|0\ 1\ \underset{\downarrow}{\underbrace{0}}\ \ldots & 1 \\ \vdots \qquad a_r & \vdots \\ \ldots|0\ 1\ 1\ 1\ 0\ \underset{\downarrow}{1}|1\ 0\ 1\ 1\ \ldots & 1 \end{array}$$

3)



In allen drei Fällen wird das erste Symmetrieelement (von rechts) mit dem Befehl 1 erreicht, wobei das Endstück eine solche Belegung aufweist, daß beim nächsten Erreichen des rechten Randes wieder einer der drei Fälle auftritt. Die Turingmaschine hält also nicht an.

Alle benutzten Computerprogramme wurden in FORTRAN geschrieben und auf einer DECsystem-10-Anlage gerechnet. Zur Ermittlung aller mit $\mathcal{R}_4$ zusammenhängenden Ergebnisse wurde insgesamt eine Rechenzeit von etwa 8 Stunden benötigt.

Zur Bestimmung von $\mathcal{R}_5$ wären Rechenzeiten notwendig, die nicht mehr vertretbar wären. Außerdem müßten neue Nichthaltekriterien entwickelt werden, da die in dieser Arbeit angegebenen angewandt auf $\mathcal{R}_5$ sicher eine nicht mehr zu bewältigende Zahl von Turingmaschinen übriglassen würden, bei denen das Nichthalten "per Hand" festgestellt werden müßte. Die Anzahl der Turingmaschinen in $\mathcal{R}_5$ ist außerdem so groß (Schätzungen ergeben sich aus Tabelle 2), daß erhebliche Speicherprobleme auftreten würden. Zu Tabelle 2 ist noch folgendes zu sagen: Die Werte $k_{0R}(5,9)$, $k_{0R}(5,10)$, $k_{1R}(5,9)$ und $k_{1R}(5,10)$ wurden deshalb nicht bestimmt, weil ab (5,9) keine Turingmaschinen mehr abgespeichert wurden. Turingmaschinen, die das Startfeld beim Ablauf nach links überschritten, wurden als nichthaltend eingestuft. Es wurde also ein einseitig begrenztes Band vorausgesetzt. Die haltenden Turingmaschinen mit 8 benutzten Teilbefehlen wurden in einem Computerprogramm durch zweimaliges Variieren eines Haltebefehls zu allen möglichen Turingmaschinen mit 9 und 10 benutzten Teilbefehlen

Universitätsbibliothek Techn. Univ. Berlin

ausgebaut. Eine Turingmaschine mit 9 benutzten Teilbefehlen wurde bei einer Gesamtschrittzahl über 250 als nichthaltend eingestuft; bei 10 benutzten Teilbefehlen war die Schranke 500. Man kann also etwa folgendes aussagen: $s_{OR}(5,10)=237$ oder $s_{OR}(5,10) \geqslant 500$.

VII Beweis der Lösbarkeit des Halteproblems bei nur schreibenden Turingmaschinen

Hopcroft und Ullman haben nachgewiesen (vgl. [9] Seite 100 Theorem 6.6), daß jede Turingmaschine durch eine Turingmaschine simuliert werden kann, die nicht löscht. Eine nicht-löschende Turingmaschine darf nur eine 1 schreiben, wenn sie auf dem Arbeitsfeld eine 1 vorfindet. Bei einer 0 auf dem Arbeitsfeld kann sie 0 oder 1 schreiben. Da das Halteproblem für Turingmaschinen nicht lösbar ist, ist mit diesem Beweis von Hopcroft und Ullman auch das Halteproblem bei nicht-löschenden Turingmaschinen nicht lösbar. Im folgenden wird nun der Beweis der Lösbarkeit des Halteproblems bei nur schreibenden Turingmaschinen erbracht.

Def. 9: Sei $\mathcal{T}_n$ die Menge aller Turingmaschinen mit n Befehlen, die nur schreiben, d.h. deren Befehle von der Form sind:

$$\begin{array}{l} i\ 0\ 1\ V_1\ n_1 \\ i\ 1\ 1\ V_2\ n_2 \end{array} \qquad V_1 \in \{R,L\}\ ;\ n_1,n_2 \in \{0,1,\ldots,n\}$$

$S^*(n)$ sei die maximal mögliche Laufzeit einer Turingmaschine aus $\mathcal{T}_n$.

Hauptsatz: Es gibt einen Algorithmus, der $S^*(n)$ für alle n bestimmt und damit das Halteproblem für nur schreibende Turingmaschinen löst.

Bemerkung: Der Beweis wird folgendermaßen geführt: Es werden zwei Kriterien hergeleitet, bei deren Erfülltsein das Nichthalten einer Turingmaschine nachgewiesen ist. Dann wird in Abhängigkeit von n eine Schranke t(n) angegeben, für die gilt: Nach Ausführung von höchstens t(n) Schritten hat jede Turingmaschine aus $\mathcal{T}_n$ entweder angehalten oder eines der beiden Kriterien erfüllt. Damit ist ein

Algorithmus gegeben, der alle haltenden Turingmaschinen aus $\mathcal{T}_n$ bestimmt, von denen man dann diejenige mit der maximalen Laufzeit heraussuchen kann.

Def. 10: Im Gegensatz zu Def. 5 in Kapitel IV soll der Zustand einer Turingmaschine vor jedem Schritt durch ein Quadrupel (i_j,a_j,l_j,r_j) beschrieben werden, wobei i_j und a_j dieselbe Bedeutung wie in Def. 5 haben. l_j und r_j werden definiert durch:

$$l_j = \min\{a_s \mid s = 0,\ldots,j-1\}$$
$$r_j = \max\{a_s \mid s = 0,\ldots,j-1\}$$

d.h., l_j ist die Nummer des bis zum (j-1)-ten Schritt einschließlich am weitesten links gelegenen Arbeitsfeldes. r_j ist das am weitesten rechts gelegene Arbeitsfeld.

Hilfssatz 1: Der Zustand von TM$\in\mathcal{T}_n$ sei (i_j,a_j,l_j,r_j), so gilt für die Bandbelegung b_j:

$$b_j(k) = \begin{cases} 0 \text{ für } k<l_j \text{ und } k>r_j \\ 1 \text{ für } l_j \leq k \leq r_j \end{cases}$$

Beweis: Da alle Felder zwischen l_j und r_j beim Ablauf von TM bis zum Zustand (i_j,a_j,l_j,r_j) bereits Arbeitsfelder waren und TM nur schreibt, sind alle diese Felder mit Einsen belegt.

Kriterium 1: Sind (i_m,a_m,l_m,r_m) und (i_k,a_k,l_k,r_k) zwei Zustände mit $m<k$, die beim Ablauf von TM$\in\mathcal{T}_n$ auftreten, und ist $i_m=i_k$, so kann in folgenden Fällen das Nichthalten von TM bewiesen werden:

a) $a_m = a_k$; $l_m = l_k$; $r_m = r_k$

b) $a_m < a_k$; $r_m = r_k+(a_k-a_m)$; $l_m = l_k$

b') $a_m > a_k$; $l_m = l_k-(a_k-a_m)$; $r_m = r_k$

Dieses Kriterium ist eine Spezialisierung des Kriteriums 1 in Ka-

pitel IV, und das Nichthalten von TM kann daher als bewiesen angesehen werden.

Hilfssatz 2: TM $\in \mathfrak{T}_n$ befinde sich im Zustand $(i_m,a_m,l_m,r_m) = z_1$. Werden von diesem Zustand ausgehend weitere $(r_m-l_m+1)\cdot n$ Schritte ausgeführt, so gilt:

a) TM hat vorher bereits angehalten;

oder b) TM hat nicht angehalten und Kriterium 1a wurde erfüllt;

oder c) TM hat nicht angehalten und Kriterium 1a wurde nicht erfüllt, so gilt für den nach Durchführung der $(r_m-l_m+1)\cdot n$ Schritte erreichten Zustand (i_k,a_k,l_k,r_k): $(l_m,r_m) \neq (l_k,r_k)$, genauer: $l_m > l_k$ oder $r_m < r_k$.

Beweis: Wird beim Ablauf von TM von z_1 aus l_m und r_m nicht verändert, sind, falls TM nicht hält, die auf z_1 folgenden Zustände von der Form (i_s,j_s,l_s,r_s) mit $l_s=l_m$ und $r_s=r_m$. Von dieser Art von Zuständen gibt es genau $(r_m-l_m+1)\cdot n$ verschiedene. a_s kann eines der r_m-l_m+1 Felder sein, die mit Einsen belegt sind, und i_s einer der Befehle 1 bis n. Nach $(r_m-l_m+1)\cdot n$ Schritten muß sich, falls TM nicht hält und r_m und l_m sich nicht ändern, ein Zustand wiederholen, womit Kriterium 1a erfüllt ist.

Satz 1: Nachdem TM $\in \mathfrak{T}_n$ vom Start an $\frac{n\cdot k\cdot(k+1)}{2}$ Schritte ausgeführt hat, gilt folgendes:

a) TM hat bereits vorher angehalten;

oder b) TM hat nicht angehalten und Kriterium 1a ist erfüllt;

oder c) TM hat nicht angehalten, Kriterium 1a ist nicht erfüllt und auf dem Band befinden sich mindestens k Einsen.

Beweis: Soll TM weder halten noch Kriterium 1a erfüllen, können nach Hilfssatz 2 bei μ vorhandenen Einsen auf dem Band höchstens $n \cdot \mu - 1$ Schritte ausgeführt werden, bevor eine weitere Eins auf das Band geschrieben wird. Unter dieser Voraussetzung werden also nach

$$\sum_{\mu=1}^{k} \mu \cdot n = \frac{n \cdot k \cdot (k+1)}{2} \text{ Schritten}$$

mindestens k Einsen auf dem Band stehen.

Hilfssatz 3: Der Zustand von TM $\in \mathcal{T}_n$ sei $z_0=(i_j,a_j,l_j,r_j)$ mit $a_j=l_j$, und der Verschiebebefehl des durchzuführenden Teilbefehls von i_j sei L. Unter der Voraussetzung, daß Kriterium 1b nicht erfüllt werden soll und die auf z_0 folgenden Zustände $z_s=(i_{j+s},a_{j+s},r_{j+s})$ s= 1,2,... sind, gibt es k mit

$$a_{j+s} = a_{j+s-1}^{-1} \quad \text{für} \quad 1 \leq s \leq k,\ k \leq n$$

$$a_{j+k+1} = a_{j+k}^{+1}$$

d.h., ohne daß der Verschiebebefehl R zwischendurch ausgeführt wird, kann höchstens n-mal der Verschiebebefehl L ausgeführt werden. (Analoges läßt sich für r_j und den Verschiebebefehl R aussagen.) Für z_k gilt außerdem

$$l_{j+k+1} = l_j - k$$

Beweis: Die Befehle $i_{j+1},i_{j+2},\ldots$ werden auf Felder mit der Belegung 0 angewandt, solange der Verschiebebefehl des durchzuführenden Teilbefehls L ist. Wiederholt sich aber ein Befehl, was nach spätestens n Schritten der Fall ist, ohne daß der Ver-

schiebebefehl R zwischendurch ausgeführt wurde, ist Kriterium 1b erfüllt.

Satz 2: Die Voraussetzungen des Hilfssatzes 3 seien erfüllt. Neben Kriterium 1b soll auch Kriterium 1a beim weiteren Ablauf von TM nicht erfüllt werden. Dann gilt:

Nach Ausführung von $n\cdot(1+n\cdot k+\frac{n(n+1)}{2})$ Schritten hat TM angehalten oder, falls der Zustand dann $z_v=(i_{j+v},a_{j+v},l_{j+v},r_{j+v})$ ist, gilt:

$$r_{j+v} \neq r_j, \quad \text{wobei } k = r_j-l_j+1 \text{ ist,}$$

d.h. spätestens nach $n\cdot(1+n\cdot k+\frac{n(n+1)}{2})$ Schritten wird der "rechte Rand" des mit Einsen belegten Bandabschnittes überschritten.

Beweis: Unter den Zuständen $z_1,z_2,\ldots,z_s,\ldots$ kann es höchstens n verschiedene geben mit $a_{j+s}=l_{j+s}-1$ und $r_{j+s}=r_j$, da es sonst zwei Zustände $(i_{j+m},a_{j+m},l_{j+m},r_{j+m})$ und $(i_{j+k},a_{j+k},l_{j+k},r_{j+k})$ gäbe mit $k>m$, $a_{j+m}=l_{j+m}-1$, $a_{j+k}=l_{j+k}-1$, $i_{j+k}=i_{j+m}$, womit Kriterium 1b erfüllt wäre. Zwischen jeweils zwei dieser maximal n Zustände können nach Hilfssatz 2 höchstens $n\cdot(r_j-l_{j+m}+1)$ Schritte erfolgen. Es können, ohne daß sich am "rechten Rand" etwas ändert, höchstens

$$\sum_{\mu=1}^{n_o} n_\mu + n\cdot(k+\sum_{\nu=1}^{\mu} n_\nu) \text{ Schritte mit } \sum_{\mu=1}^{n_o} n_\mu \leq n$$

erfolgen. Erst kommen n_1 Einsen links dazu, dann führt TM maximal $(k+n_1)\cdot n$ Schritte aus, ohne daß sich an der Zahl der Einsen etwas ändert (Hilfssatz 2), dann kommen n_2 Einsen links dazu und TM kann anschließend ohne Änderung der Zahl der Einsen $(k+n_1+n_2)\cdot n$ Schritte ausführen usw. Dies läßt sich ohne Veränderung des rechten Randes höchstens solange fortsetzen, wie $\sum_{\mu=1}^{n_o} n_\mu=n$ ist. Die

obige Summe läßt sich, wie leicht einzusehen ist, abschätzen, indem man $n_\mu = 1$ für alle μ und $n_0=n$ setzt.

Also:

$$\sum_{\mu=1}^{n_0} n_\mu+n\cdot(k+\sum_{\nu=1}^{\mu} n_\nu) = \sum_{\mu=1}^{n} (1+n\cdot(k+\mu)) = n(1+n\cdot k+\frac{n(n+1)}{2})$$

Bemerkung: Verläßt der Lese- und Schreibkopf nach Erreichen des Zustandes $z_v=(i_v,a_v,l_v,r_v)$ innerhalb der nächsten n Schritte den Bandabschnitt zwischen l_v und r_v nicht, so gibt es einen Zustand $z_w=(i_w,a_w,l_w,r_w)$ $w>v$ und eine ganze Zahl $s\leq n$ mit

$$i_v = i_w \quad \text{und} \quad a_w=a_v+s \quad \text{oder} \quad a_w=a_v-s,$$

d.h. der Ablauf zwischen z_v und z_w stellt eine Schleife dar, die solange wiederholt wird, wie der Bandabschnitt zwischen l_v und r_v nicht verlassen wird. Bei einmaligem Durchlaufen der Schleife wird das Arbeitsfeld um s Felder nach rechts bzw. nach links verschoben.

Hilfssatz 4: $TM\in\mathcal{T}_n$ sei im Zustand $z_v=(i_v,a_v,l_v,r_v)$ mit

$$l_j+n\leq a_j \quad \text{und} \quad r_j-n\geq a_j$$

so befindet sich TM in einer Schleife.

Beweis: TM hat sich beim Erreichen des Zustandes z_v mehr als n Schritte im Bandabschnitt zwischen l_j und r_j aufgehalten.

Kriterium 2: $TM\in\mathcal{T}_n$ erreiche beim Ablauf folgende zwei Zustände:

$z_j = (i_j,a_j,l_j,r_j)$ und $z_k = (i_k,a_k,l_k,r_k)$ mit $k>j$, für die gelten:

a) $a_j = l_j$; $a_k = l_k$

b) $i_j = i_k$

c) $r_j - l_j + 1 \geq 2n$, $r_k-l_k \equiv r_j-l_j \bmod n!$
$(r_k-l_k = r_j-l_j+w\cdot n!)$

so hält TM nicht an.

Beweis: z_j kann durch folgende Operation in z_k übergeführt werden. Es gibt ein Feld f, für das gilt:

$$f-l_j \geq n-1 \quad \text{und} \quad r_j-f \geq n-1$$

Zwischen f und f+1 werden w·n! Felder mit der Belegung 1 eingefügt. Numeriert man dann die Felder entsprechend um, wird damit z_j in z_k übergeführt.

Behauptung: Diese Einfügung hat keine Bedeutung für das Halten oder Nichthalten von TM. Nach Hilfssatz 4 befindet sich wegen c) TM, wenn bei zwei aufeinanderfolgenden Schritten das Arbeitsfeld von f nach f+1 bzw. umgekehrt wechselt, in einer Schleife, die den Lese- und Schreibkopf durch einmaliges Durchlaufen um eine Anzahl Felder, die höchstens n betragen kann, nach rechts oder links verschiebt. Da zwischen f und f+1 w·n! Felder mit Einsen eingeschoben werden, wird der Befehl, der durchgeführt wird, wenn TM f+1 bzw. f erreicht, unabhängig davon sein, ob zwischen f und f+1 ein Vielfaches von n! Feldern mit Einsen eingeschoben wurde oder nicht.

Man kann umgekehrt beim Zustand z_k w·n! Felder herausnehmen und hat wieder den Zustand z_j. Dies kann man beliebig wiederholen, so daß TM also nicht hält.

Beweis des Hauptsatzes:

Mit $t(n) = n+3n^2+2n^3+2n^2 \cdot n!+5n^3 \cdot n!+2n^4(n!)^2$

und Kriterium 1 und 2 ist der Hauptsatz bewiesen.

Beweis: $TM \in \mathcal{T}_n$ halte nicht und erfülle beim Ablauf nicht Kriterium 1, so gilt wegen Satz 1, nach $\frac{n\cdot(2n+1)(2n+2)}{2}=2n^3+3n^2+n$ Schritten stehen mindestens 2n+1 Einsen auf dem Band. Daraus folgt, daß TM vorher einen Zustand $z_{j_1}=(i_{j_1},a_{j_1},l_{j_1},r_{j_1})$ mit $a_{j_1}=l_{j_1}$ bzw. $a_{j_1}=r_{j_1}, r_{j_1}-l_{j_1}+1=2n$ und durchzuführenden Verschiebebefehl L bzw. R

erreicht hat. $\{z_{i_s}\}_{s=2,3,\ldots}$ sind die im weiteren Ablauf von TM irgendwann auftretenden Zustände (wegen Satz 1 gesichert) mit $a_{i_s}=l_{i_s}$ bzw. $a_{i_s}=r_{i_s}$ und durchzuführender Verschiebebefehl ist L bzw. R. Von dieser Art von Zuständen kann es maximal $2n!\cdot n$ (für jede Seite $n!\cdot n$) verschiedene geben, ohne daß Kriterium 2 erfüllt wäre. Für eine Seite: n Befehle, n! mögliche Anzahlen von Einsen, die modulo n! verschieden sind. Nach maximal $n+(k+n)\cdot n$ Schritten, $k=r_{j_s}-l_{j_s}+1$, wird, ausgehend von einem dieser $2\cdot n!\cdot n$ Zustände, der nächste erreicht. n Einsen können nach Hilfssatz 3 maximal hinzukommen, und anschließend kann sich der Lese- und Schreibkopf maximal $(k+n)\cdot n$ Schritte im Feld der Einsen aufhalten (Hilfssatz 2), ohne daß Kriterium 1 erfüllt ist. Vom Zustand z_{j_1} ($r_{j_1}-l_{j_1}+1=2n$) bis zum Zustand $z_{j_{2n!\cdot n}}$ (spätestens dort ist Kriterium 2 erfüllt) können maximal $\sum_{\mu=1}^{2n!\cdot n} n+(2n+\mu\cdot n)\cdot n$ Schritte ausgeführt werden, ohne daß Kriterium 2 erfüllt ist. Insgesamt ist also bei einer nichthaltenden Turingmaschine aus $\mathcal{T}_n$ nach spätestens

$$\sum_{u=1}^{2n\cdot n!} (n+n(2n+\mu n))+n+3n^2+2n^3 = n+3n^2+2n^3+2n^2\cdot n!+5n^3\cdot n! +2n^4\cdot(n!)^2 = t(n)$$

Kriterium 1 oder 2 erfüllt.

Bemerkung: Die maximal mögliche Laufzeit in $\mathcal{T}_4$ ist 31 und die maximal mögliche Feldweite (gleich der maximal möglichen Anzahl Einsen auf dem Band nach dem Halt) ist 9. Beide Werte werden z.B. durch folgende Turingmaschine realisiert:

1	2	3	4	σ	φ	β	d_1/d_2
101R2	201R3	301L1	401R0	31	9		
111L1	211L2	311R4	411R3		(-3,5)	9	15/31

Literaturverzeichnis

1) Turing, A.,M. "On computable numbers, with an application to the Entscheidungsproblem." Proceedings of the London Mathematical Society (2), 42 (1936-1937), 230-265

2) Kleene, S.,C. "Introduction to Metamathematics." D. Van Nostrand Co., Princeton, New Jersey, 1952.

3) Rado, T. "On non-computable functions." Bell System Technical Journal, 41, 3 (1962), 877-884.

4) Rado, T. "On a sample source for non-computable functions." Proceedings of the Symposium on Mathematical Theory of Automata, Brooklyn, New-York, 1963, 75-81.

5) Lin, S., Rado, T. "Computer Studies of Turing Machine Problems." Journal of the ACM, 12, 2 (1965), 196-212.

6) Green, M.,W. "A Lower Bound on Rado's Sigma Function for Binary Turing Machines." Proceedings of the 5th Annual Symposium on Switching Circuits Theory and Logical Design, Princeton, New Jersey, 1964, 91-94.

7) Korfhage, R.,R. "Logic and Algorithms." John Wiley & Sons, Inc., New-York, London,Sydney, 1966.

8) Brauer, W., Indermark, K. "Algorithmen, rekursive Funktionen und formale Sprachen." B·I-Hochschulskripten, 817*, Bibliographisches Institut Mannheim, 1968.

9) Hopcroft, J.,E., Ullman, J.,D. "Formal Languages and their Relation to Automata." Addison-Wesley Publishing Company, Massachusetts, 1969.

Lebenslauf

Am 12. Juli 1942 wurde ich als Sohn des Oberingenieurs Karl Weimann und seiner Ehefrau Gertrud, geb. Hannig, in Mülheim-Ruhr geboren.

Nach dem Besuch der Volksschule in Mülheim-Ruhr (1949-1953) wechselte ich Ostern 1953 zum Staatlichen Gymnasium in Mülheim-Ruhr, wo ich im Februar 1963 die Reifeprüfung bestand.

Zum Sommersemester 1956 begann ich das Studium der Mathematik an der Rheinischen Friedrich-Wilhelms-Universität Bonn. Die Diplomhauptprüfung legte ich im Oktober 1968 ab, nachdem ich zuvor unter Leitung von Herrn Prof. Dr. H. Schiek meine Diplomarbeit mit dem Thema "Theorie der modularen Charaktere und ihre Anwendung auf einfache Gruppen" angefertigt hatte.

Seit November 1968 bin ich im Rechenzentrum des Max-Planck-Instituts für Kohlenforschung in Mülheim-Ruhr beschäftigt.

Am 11. April 1969 heiratete ich die Krankenschwester Rosemarie Mahl. Unser Sohn Boris wurde am 1. August 1970 und unsere Tochter Tatjana am 8. September 1972 geboren.

Vom Beginn des Jahres 1971 an bin ich nach Anregung durch eine Vorlesung von Herrn Prof. Dr. W. Brauer mit der Ausführung der vorliegenden Arbeit beschäftigt.