

### 7.3.2 Fleißige Biber

Im Jahre 1961 führte der ungarische Mathematiker Tibor Radó das von ihm so bezeichnete »Busy Beaver Game« ein.<sup>1</sup>

**Definition 7.1**  
Fleißiger Biber  
busy beaver

Ein *fleißiger Biber* ist eine deterministische Einband-Turing-Maschine mit dem Bandalphabet  $\{0, 1\}$  und dem Leerzeichen  $\emptyset$ , die den Kopf bei jedem Zustandsübergang entweder nach links oder rechts bewegt, nicht aber auf der aktuellen Position belässt. Jeder fleißige Biber verfügt über  $n$  innere Zustände  $z_1, \dots, z_n$ , in denen er nicht hält, und einen zusätzlichen Endzustand  $z_0$ . Ohne Beschränkung der Allgemeinheit sei  $z_1$  der Startzustand der Maschine. Anfangs sind sämtliche Felder des nach beiden Seiten ausgedehnten Bandes mit dem Leerzeichen  $\emptyset$  belegt, so dass die leere Menge an die Stelle des Eingabealphabets tritt. Gemäß unserer zuvor eingeführten Notation hat ein solcher fleißiger Biber also die Form

$$M = (\{z_0, z_1, \dots, z_n\}, \{0, 1\}, \delta, z_1, \{z_0\}, \emptyset) \quad \text{mit} \\ \delta : \{z_1, \dots, z_n\} \times \{0, 1\} \rightarrow \{z_0, z_1, \dots, z_n\} \times \{0, 1\} \times \{L, R\}. \quad \blacksquare$$

Im Folgenden zählen wir den obligatorischen Endzustand  $z_0$  nicht mit. Wenn wir verkürzend von  $n$  Zuständen sprechen, meinen wir damit stets die inneren Zustände  $\{z_1, \dots, z_n\}$ .

Radó stellt den Lesern seiner Publikation die Aufgabe, eine beliebige Anzahl an Zuständen auszuwählen und einen fleißigen Biber zu konstruieren, der *möglichst viele Einsen* auf das Band schreibt. Diese müssen nicht zwangsläufig zusammenhängend auf dem Band erscheinen; sie dürfen auch durch beliebig viele Nullen getrennt sein. Die Maschine muss aber nach *endlich vielen Schritten* halten, darf also nicht in eine Endlosschleife geraten.

Die Teilnehmer an dem von Radó ausgeschriebenem Wettbewerb müssen angeben, nach wie vielen Schritten  $s$  ihre Maschine  $M$  im Endzustand  $z_0$  hält, sie müssen also das Paar  $(M, s)$  einreichen. Aus diesem Grund ist die Behauptung über die Anzahl an geschriebenen Einsen leicht nachzuprüfen. Man muss  $M$  lediglich im Startzustand  $z_1$  anwerfen und maximal  $s$  Schritte ausführen lassen. Hält die Maschine nach weniger als  $s$  Schritten, so wird sie dem Teilnehmer zur Korrektur zurückgegeben. Hat sie bei der Überprüfung nach  $s$  Schritten noch nicht gehalten, wird sie disqualifiziert. Erfüllt sie aber die von Radó aufgestellten Bedingungen, so gilt die Anzahl an Einsen, die sie auf dem Band hinterlässt, hier mit  $\sigma(M, s)$  bezeichnet, als die im Wettbewerb erreichte Punktzahl.

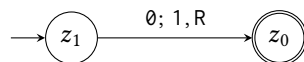
Biber sind vor allem für ihre teils gigantischen Dammbauten bekannt, mit denen sie Bäche aufstauen.

<sup>1</sup> T. Radó: *On non-computable functions*. The Bell System Technical Journal, Band 41, Nr. 3, 1962, S. 877–884.

### 7.3 Theoretische Grenzen

Bevor wir uns näher damit beschäftigen, zu welchem Zweck Radó das Busy Beaver Game ins Leben rief, sehen wir uns zunächst einige Turing-Maschinen, die mit Fug und Recht als fleißige Biber bezeichnet werden dürfen, etwas genauer an.

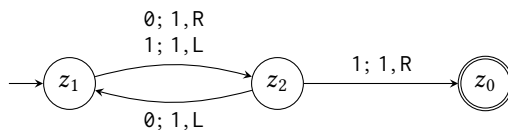
Offensichtlich ist die Turing-Maschine  $M_1$ , die durch den folgenden Graphen gegeben ist, ein fleißiger Biber mit einem Zustand. Startet man  $M_1$  im Zustand  $z_1$ , so schreibt diese Maschine in einem Schritt genau eine Eins auf das Band und hält anschließend im Endzustand  $z_0$ .



**Abbildung 7.1**  
Fleißiger Biber  $M_1$   
mit einem Zustand

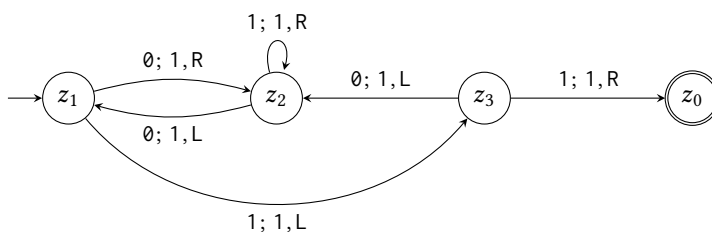
Es leuchtet sofort ein, dass ein fleißiger Biber mit einem Zustand nicht mehr als eine Eins schreiben kann, ohne direkt nach dem Start in eine Endlosschleife zu geraten. Statt nach rechts könnte sich der Kopf beim Übergang von  $z_1$  in  $z_0$  auch nach links bewegen, was aber auf den Fleiß der Maschine keinerlei Auswirkung hätte.

Die Turing-Maschine  $M_2$  mit zwei Zuständen hinterlässt in sechs Schritten immerhin schon vier zusammenhängende Einsen auf dem Band.



**Abbildung 7.2**  
Fleißiger Biber  $M_2$   
mit zwei Zuständen

Die folgende Abbildung zeigt den Graphen des fleißigen Bibers  $M_3$ , der in dreizehn Schritten sechs zusammenhängende Einsen generiert und dabei die in Tabelle 7.2 aufgelisteten Konfigurationen durchläuft.



**Abbildung 7.3**  
Fleißiger Biber  $M_3$   
mit drei Zuständen

Spätestens jetzt stellt sich die Frage, ob  $M_3$  der fleißigste aller Biber mit drei Zuständen ist, oder ob sich die Zahl der Einsen noch steigern ließe. Aufgrund des scheinbar chaotischen Verhaltens von  $M_3$  ist diese Frage alles andere als leicht zu beantworten.

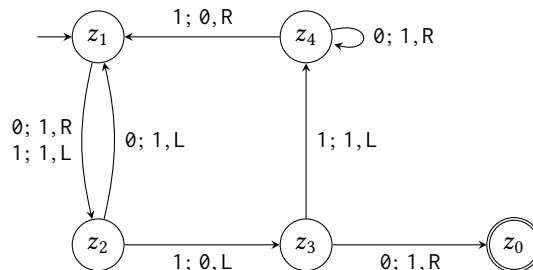
7 Berechenbarkeit

**Tabelle 7.2**  
Konfigurationen,  
welche der fleißige  
Biber  $M_3$  gemäß Ab-  
bildung 7.3 durchläuft

Schritt	Zustand	Bandinhalt und Kopfposition
	$z_1$	. . . 0 0 0 0 <u>0</u> 0 0 0 0 . . .
1	$z_2$	. . . 0 0 0 0 1 <u>0</u> 0 0 0 . . .
2	$z_2$	. . . 0 0 0 0 <u>1</u> 1 0 0 0 . . .
3	$z_3$	. . . 0 0 0 <u>0</u> 1 1 0 0 0 . . .
4	$z_2$	. . . 0 0 <u>0</u> 1 1 1 0 0 0 . . .
5	$z_1$	. . . 0 <u>0</u> 1 1 1 1 0 0 0 . . .
6	$z_2$	. . . 0 1 <u>1</u> 1 1 1 0 0 0 . . .
7	$z_2$	. . . 0 1 1 <u>1</u> 1 1 0 0 0 . . .
8	$z_2$	. . . 0 1 1 1 <u>1</u> 1 0 0 0 . . .
9	$z_2$	. . . 0 1 1 1 1 <u>1</u> 0 0 0 . . .
10	$z_2$	. . . 0 1 1 1 1 1 <u>0</u> 0 0 . . .
11	$z_2$	. . . 0 1 1 1 1 1 <u>1</u> 0 0 . . .
12	$z_3$	. . . 0 1 1 1 <u>1</u> 1 1 0 0 . . .
13	$z_0$	. . . 0 1 1 1 1 <u>1</u> 1 0 0 . . .

Etwa zehn Jahre nach Ausschreibung des Wettbewerbs fanden Weimann, Casper und Fenzel den fleißigen Biber  $M_4$ , der in 107 Schritten 13 Einsen der Form  $1011111111111$  auf das anfangs leere Band schreibt.<sup>2</sup>

**Abbildung 7.4**  
Fleißiger Biber  $M_4$   
mit vier Zuständen

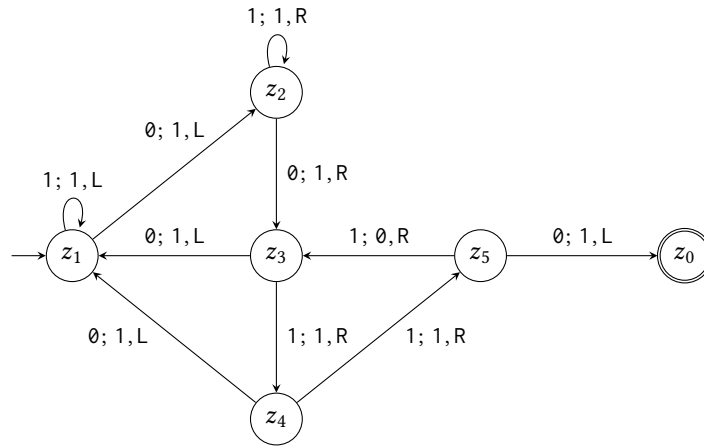


Der geneigte Leser möge sich mittels Papier und Bleistift oder einer Simulation gern ansehen, wie die Einsen nach und nach auf dem Band erscheinen. Wenngleich der Vorgang »vollkommen« deterministisch ist, so lässt sich kaum ein »System« erkennen, nach dem die Turing-Maschine arbeitet. Schon bei vier Zuständen scheint es schier unmöglich zu sein, ein geeignetes Konstruktionsprinzip für einen fleißigen Biber herauszuarbeiten, das die Chancen auf eine gute Wettbewerbsplatzierung garantiert.

<sup>2</sup> B. Weimann, K. Casper, W. Frenzel: *Untersuchungen über haltende Programme für Turingmaschinen mit 2 Zeichen und bis zu 5 Befehlen*. GI Gesellschaft für Informatik e. V.: 2. Jahrestagung, Karlsruhe, Germany, 1972. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, 1973, S. 72–81.

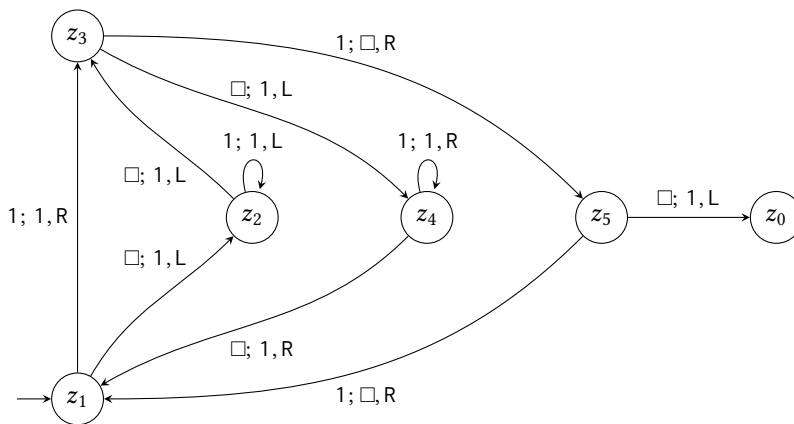
### 7.3 Theoretische Grenzen

Im Jahre 1989 veröffentlichten Heiner Marxen und Jürgen Buntrock den fleißigen Biber  $M_5$ . Diese Maschine schreibt in 11 798 826 Schritten stolze 4098 Einsen auf das Band.<sup>3</sup>



**Abbildung 7.5**  
Fleißiger Biber  $M_5$   
mit fünf Zuständen

Die Turing-Maschine  $M'_5$ , die ebenfalls von Marxen und Buntrock veröffentlicht wurde, hält erst nach 47 176 870 Schritten, hinterlässt aber die gleiche Anzahl an Einsen wie  $M_5$ .<sup>4</sup>



**Abbildung 7.6**  
Fleißiger Biber  $M'_5$   
mit fünf Zuständen

Die Maschinen  $M_5$  und  $M'_5$  sind die aktuellen Rekordhalter für die Anzahl an Einsen in der Klasse mit fünf Zuständen. Sind sie aber auch die fleißigsten Biber, oder gibt es eine Turing-Maschine, welche die Anforderungen von Radó erfüllt und mehr als 4098 Einsen auf das Band schreibt? Und wie

<sup>3</sup> H. Marxen, J. Buntrock: *Attacking the Busy Beaver 5*. Bulletin of the EATCS, Nr. 40, 1990, S. 247–251.

<sup>4</sup> Ebd.

7 Berechenbarkeit

**Tabelle 7.3**  
Anzahl ausgeführter Schritte und Anzahl geschriebener Einsen für fleißige Biber mit ein bis fünf Zuständen

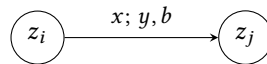
Anzahl an Zuständen $n$	Fleißiger Biber $M$	Anzahl an Schritten $s$	Anzahl an Einsen $\sigma(M, s)$	Autoren
1	$M_1$	1	1	Radó
2	$M_2$	6	4	Radó
3	$M_3$	13	6	Lin, Radó
4	$M_4$	107	13	Weimann, Casper, Fenzel
5	$M_5$	11 798 826	4098	Marxen, Buntrock
	$M'_5$	47 176 870	4098	Marxen, Buntrock

viele Schritte kann ein fleißiger Biber maximal ausführen, bevor dieser den Endzustand  $z_0$  erreicht? In der Klasse mit fünf Zuständen führt die Maschine  $M'_5$  derzeit auch diese Rangliste an. Ist es möglich, einen fleißigen Biber zu finden, der  $M'_5$  in seiner enormen Ausdauer übertrifft?

Tabelle 7.3 fasst die aktuellen Rekorde zusammen. Wie lange aber werden diese Bestand haben? Kann man für eine gegebene Anzahl  $n$  an Zuständen die Maximalzahl  $\Sigma(n)$  an Einsen und die Maximalzahl  $S(n)$  an Schritten bestimmen, um einem Kandidaten gegebenenfalls den unverrückbaren ersten Platz zusprechen zu können?

Zur Beantwortung der letztgenannten Frage mag es zunächst naheliegender scheinen, alle möglichen Turing-Maschinen mit  $n$  Zuständen zu generieren und systematisch zu untersuchen. Dazu betrachten wir eine beliebige Transition der Form

**Abbildung 7.7**  
Beliebige Transition eines fleißigen Bibers



beziehungsweise, in der funktionalen Schreibweise,

$$(7.1) \quad \delta(z_i, x) = (z_j, y, b).$$

Da der Endzustand  $z_0$ , sobald er einmal erreicht wurde, nicht mehr verlassen werden darf, gilt  $i = 1, \dots, n$ . Die Variable  $x$  steht für ein beliebiges Eingabezeichen, es gilt also  $x \in \{0, 1\}$ . Somit gibt es  $2n$  verschiedene Paare  $(z_i, x)$ , welche als Argument der Überföhrungsfunktion auftreten können. Wegen  $j = 0, \dots, n$ ,  $y \in \{0, 1\}$  und  $b \in \{L, R\}$  gibt es  $(n + 1) \cdot 2 \cdot 2 = 4n + 4$  verschiedene Tripel  $(z_j, y, b)$  als mögliche Werte der Überföhrungsfunktion.

Die Funktion  $\vartheta : \mathbb{N} \rightarrow \mathbb{N}$  ordnet der Anzahl an Zuständen die Anzahl an möglichen Turing-Maschinen zu. Nach den Regeln der Kombinatorik gilt

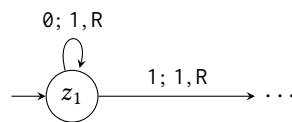
$$(7.2) \quad \vartheta(n) = (4n + 4)^{2n}.$$

Anzahl an Zuständen $n$	Anzahl an Turing-Maschinen mit Überföhrungsfunktion $\delta : \{z_1, \dots, z_n\} \times \{0, 1\} \rightarrow \{z_0, \dots, z_n\} \times \{0, 1\} \times \{L, R\}$ $\vartheta(n)$
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
7	1 180 591 620 717 411 303 424
8	7 958 661 109 946 400 884 391 936
9	68 719 476 736 000 000 000 000 000
10	739 696 442 014 594 807 059 393 047 166 976

**Tabelle 7.4**  
Anzahl möglicher Turing-Maschinen in Abhängigkeit von der Anzahl an Zuständen

Tabelle 7.4 zeigt, dass  $\vartheta$  unglücklicherweise rasant mit  $n$  wächst. Würden wir uns mit jeder Turing-Maschine, die fünf Zustände hat, auch nur eine Sekunde lang beschäftigen, so nähme dies bereits über zwei Millionen Jahre in Anspruch.

Selbstverständlich befinden sich unter den  $\vartheta(n)$  Turing-Maschinen höchst seltsame Exemplare, die wir teilweise von geeigneten Algorithmen aussondern lassen könnten. Dazu zählen solche Maschinen, bei denen mindestens ein Zustand von  $z_1$  aus nicht erreichbar ist oder die wie beispielsweise



**Abbildung 7.8**  
Turing-Maschine, die, auf leerem Band gestartet, in eine Endlosschleife gerät

unmittelbar nach ihrem Start in eine Endlosschleife geraten. Darüber hinaus könnten wir es mit etwas Aufwand vermeiden, Turing-Maschinen zu generieren, die bis auf die Nummerierung der Zustände identisch sind. All dies änderte aber nichts an der Tatsache, dass eine Brute-Force-Analyse für größere  $n$  in der Praxis völlig unrealistisch ist.

Ganz umsonst waren unsere Überlegungen nicht, denn immerhin können wir nun feststellen, dass die Anzahl der fleißigen Biber mit  $n$  Zuständen, die wir im Folgenden  $\beta(n)$  nennen, auf jeden Fall kleiner als  $\vartheta(n)$  ist. Andererseits gibt es für jedes  $n$  mindestens zwei fleißige Biber, wie die Graphen in Abbildung 7.9 zweifelsfrei belegen. Zusammengefasst gilt also für die Anzahl an fleißigen Bibern mit  $n$  Zuständen

$$1 < \beta(n) < (4n + 4)^{2n} . \tag{7.3}$$

7 Berechenbarkeit

**Abbildung 7.9**  
 Fleißige Biber, die  
 eine Eins schreiben  
 und dann halten



An dieser Stelle widerstehen wir der Versuchung, nach einem Funktionsterm für die Funktion  $\beta : \mathbb{N} \rightarrow \mathbb{N}$  zu forschen, die jeder Anzahl an Zuständen die Anzahl an fleißigen Bibern zuordnet. Stattdessen definieren wir die Menge  $B_n$ , die alle Paare  $(M, s)$  enthält, für die  $M$  ein fleißiger Biber mit  $n$  Zuständen ist, der nach  $s$  Schritten im Endzustand hält. Die Menge  $B_n$  enthält genau  $\beta(n)$  Elemente, ist also nicht leer und endlich. Wie eingangs erwähnt, können wir für jeden Wettbewerbskandidaten  $(M, s)$  durch Simulation problemlos feststellen, ob  $(M, s) \in B_n$  gilt.

Für einen Wettbewerbskandidaten  $(M, s) \in B_n$  haben wir die Anzahl an Einsen, die der fleißige Biber  $M$  in  $s$  Schritten auf das Band schreibt, mit  $\sigma(M, s)$  bezeichnet. Für ein gegebenes  $n$  ist die Menge aller Punktezahlen

$$\{\sigma(M, s) \mid (M, s) \in B_n\}$$

ebenfalls wohldefiniert, nicht leer und endlich. Nun begeben wir uns gedanklich auf die Suche nach dem fleißigsten und dem ausdauerndsten Biber mit  $n$  Zuständen. Dazu führen wir die Funktionen  $\Sigma : \mathbb{N} \rightarrow \mathbb{N}$  und  $S : \mathbb{N} \rightarrow \mathbb{N}$  ein, welche jeder Anzahl  $n$  an Zuständen die Maximalzahl an Einsen beziehungsweise die Maximalzahl an Schritten zuordnen, die formal also durch

$$(7.4) \quad \Sigma(n) := \max_{(M,s) \in B_n} (\sigma(M, s)) \quad \text{und} \quad S(n) := \max_{(M,s) \in B_n} (s)$$

definiert sind.

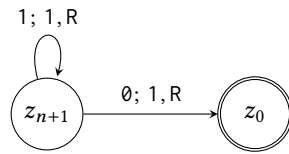
**Satz 7.1**  
 Strenge Monotonie  
 der Radó-Funktion

Wir können uns mühelos davon überzeugen, dass die Funktion  $\Sigma$  streng monoton wächst, dass also

$$\Sigma(n + 1) > \Sigma(n) \quad \text{für alle} \quad n \geq 1. \quad \blacksquare$$

*Beweis*

Dazu nehmen wir an, wir hätten den fleißigsten Biber mit  $n$  Zuständen gefunden. Nun konstruieren wir einen fleißigen Biber mit  $n + 1$  Zuständen, indem wir den bisherigen Endzustand  $z_0$  durch den neuen Zustand  $z_{n+1}$  ersetzen und dafür sorgen, dass die Maschine in willkürlich gewählter Richtung alle eventuell vorhandenen Einsen überliest, eine weitere Eins auf das erste leere Feld schreibt und dabei in den Endzustand wechselt, wie dies in Abbildung 7.10 gezeigt ist. Demzufolge produziert die neue Maschine mit  $n + 1$  Zuständen genau eine Eins mehr als der fleißigste Biber mit  $n$  Zuständen. ■



**Abbildung 7.10**  
Erweiterung eines fleißigen Bibers um einen Zustand

Die Überlegungen für  $\Sigma$  können wir ohne Weiteres auf die Funktion  $S$  übertragen. Auch  $S$  ist streng monoton, daß heißt,

$$S(n + 1) > S(n) \quad \text{für alle } n \geq 1. \quad \blacksquare$$

Wenden wir die Konstruktion gemäß Abbildung 7.10 an, so erkennen wir nämlich, dass der fleißige Biber mit  $n + 1$  Zuständen mindestens einen Schritt mehr ausführt als derjenige mit  $n$  Zuständen.  $\blacksquare$

Bis heute ist es gerade einmal bis zum Wert  $n = 4$  mit Hilfe teils sehr aufwändiger Analysen gelungen, die Funktionswerte von  $\Sigma$  und  $S$  exakt zu bestimmen. Dabei hat sich gezeigt, dass die in Tabelle 7.3 für die fleißigen Biber  $M_1$  bis  $M_4$  angegebenen Werte  $s$  und  $\sigma(M, s)$  tatsächlich die Maximalzahlen  $S(n)$  und  $\Sigma(n)$  darstellen. Schon für  $n = 5$  sind die Funktionswerte von  $\Sigma$  und  $S$  aber bis heute unbekannt.

Durch Abschätzungen kann man sich davon überzeugen, dass  $\Sigma$  und  $S$  *sehr stark* wachsen. So erwähnt Radó beispielsweise, dass man eine Turing-Maschine, welche zunächst sieben Einsen auf das leere Band schreibt und viermal hintereinander die Fakultät in Unärdarstellung über dem Bandalphabet  $\{0, 1\}$  berechnet, mit höchstens 100 Zuständen realisieren kann. Somit muss  $\Sigma(100)$  mindestens den Wert  $((((7!)!)!)!$  haben,<sup>5</sup> was eine so gigantische Zahl ist, dass handelsübliche Digitalrechner weder die Dezimaldarstellung noch einen Näherungswert dafür bestimmen können.

Radó zeigt in seiner Veröffentlichung aus dem Jahr 1962 weiterhin, dass die Funktionen  $\Sigma$  und  $S$  stärker als jede Turing-berechenbare Funktion wachsen und daher nicht berechenbar sind.<sup>6</sup>  $\blacksquare$

Nachdem wir im vorangegangenen Abschnitt einsehen mussten, dass es Funktionen gibt, die nicht berechenbar sind, haben wir mit  $\Sigma$  und  $S$  nun zwei konkrete Vertreter dieser Spezies kennengelernt. Obwohl  $\Sigma$  und  $S$  mathematisch wohldefiniert sind, gibt es keine Turing-Maschine, welche die entsprechenden Funktionswerte berechnen könnte. Es sieht also danach aus, als würde die mühsame Suche nach fleißigen Bibern niemals enden.

**Satz 7.2**  
Strenge Monotonie der Funktion  $S$

*Beweis*

**Satz 7.3**  
Nicht-Berechenbarkeit von  $\Sigma$  und  $S$

Wir verzichten hier auf die Wiedergabe der Beweise, da diese recht »technisch« ausfallen.

<sup>5</sup> T. Radó: *On non-computable functions*. The Bell System Technical Journal, Band 41, Nr. 3, 1962, S. 877–884.

<sup>6</sup> Ebd.