universität
wien

BACHELORARBEIT

Titel der Bachelorarbeit

Ordinalzahl-Turingmaschinen

verfasst von

Lena Birschitzky

angestrebter akademischer Grad

Bachelor of Science (BSc.)

Wien, im September 2020

Studienkennzahl lt. Studienblatt:	A 033621
Studienrichtung lt. Studienblatt:	Mathematik
Betreuerin:	Dr. Sandra Müller

Abstract

Ordinalzahl-Turingmaschinen (infinite time turing machines) sind ein Modell der theoretischen Mathematik für Berechnungen mit unendlich vielen Schritten. In dieser Arbeit werden zunächst Grundlagen über Ordinalzahlen und Turingmaschinen geklärt, um anschließend die Funktionsweise von Ordinalzahl-Turingmaschinen beschreiben zu können. Die Begriffe *berechenbar* und *entscheidbar* für solche Maschinen werden eingeführt und untersucht. Es zeigt sich zum Beispiel, dass jede $\mathbf{\Pi}^1_1$-Menge in diesem Sinn entscheidbar ist.

Inhaltsverzeichnis

1 Einleitung

Diese Arbeit beschäftigt sich mit Ordinalzahl-Turingmaschinen, einem Konzept aus der theoretischen Informatik, das von Hamkins und Lewis [2] erstmals beschrieben wurde. Es handelt sich dabei um eine Erweiterung der gewöhnlichen Turingmaschinen, mit der Berechnungen mit unendlich vielen Rechenschritten beschrieben werden können.
In Abschnitt 2 werden Ordinalzahlen eingeführt und einige grundlegende Eigenschaften gezeigt. Diese Inhalte sind in den meisten einführenden Büchern über Mengenlehre zu finden, diese Arbeit orientiert sich dabei an [4] und [7].
Abschnitt 3 ist eine kurze Einführung in die Theorie der Turingmaschinen, die ungefähr Kapitel 5 von Hedtstück [3] folgt.
Abschnitt 4 behandelt schließlich Ordinalzahl-Turingmaschinen und beruht hauptsächlich auf [2]. Zunächst wird die Arbeitsweise solcher Maschinen beschrieben; dabei werden Ordinalzahlen zur Nummerierung der Rechenschritte verwendet. Der Begriff *in unendlicher Zeit entscheidbar* wird analog zur Entscheidbarkeit bei gewöhnlichen Turingmaschinen definiert. Anschließend wird untersucht, welche Mengen in unendlicher Zeit entscheidbar sind, was gewissermaßen zeigt, wie „mächtig“ Ordinalzahl-Turingmaschinen sind.
Zu diesem Zweck gibt es einen kleinen Exkurs in die Grundlagen der deskriptiven Mengenlehre, der schließlich zu dem Ergebnis führt, dass jede $\mathbf{\Pi}^1_1$-Menge in unendlicher Zeit entscheidbar ist. Hier werden hauptsächlich Kechris [6], Kanamori [5] und Schindler [8] verwendet, wobei die Notation großteils von [8] übernommen wird.
In dieser Arbeit werden ein paar Grundkenntnisse über mathematische Logik vorausgesetzt, insbesondere über aussagenlogische Formeln sowie über Mengenlehre und die ZFC-Axiome. Davon abgesehen werden aber alle benötigten Grundlagen in den Abschnitten 2 und 3 wiederholt, sodass die Arbeit auch ohne spezielles Vorwissen nachvollziehbar ist.

2 Ordinalzahlen

In diesem Abschnitt werden Ordinalzahlen definiert und untersucht. Es wird festgestellt, dass jede natürliche Zahl eine Ordinalzahl ist, und dass die Ordinalzahlen über die natürlichen Zahlen hinausgehen. Außerdem zeigt sich, dass die Ordinalzahlen durch die Relation $\in$ wohlgeordnet sind, und dadurch dazu geeignet sind, die Position von Elementen in Folgen anzugeben.

Definition. Die *von Neumann-Zahlen* sind rekursiv wie folgt definiert:

$$\begin{aligned} 0 &:= \emptyset \\ 1 &:= \{\emptyset\} = \{0\} \\ 2 &:= \{\emptyset, \{\emptyset\}\} = \{0, 1\} \\ &\dots \\ n+1 &:= n \cup \{n\} = \{0, 1, ..., n\} \end{aligned}$$

Die Menge aller von Neumann-Zahlen wird mit ω bezeichnet.

Bemerkung. Auf den von Neumann-Zahlen lassen sich Operationen $+$, $\cdot$ und *exp* definieren, die gemeinsam mit $\in$ die Peano Arithmetik erfüllen. Sie stellen also ein Modell der natürlichen Zahlen dar. Im Folgenden werden Ordinalzahlen betrachtet, die man als Verallgemeinerung der so konstruierten natürlichen Zahlen verstehen kann.

Definition. Eine Menge x heißt *transitiv*, falls für alle $y \in x$ auch gilt $y \subseteq x$.

Definition. Eine Menge x heißt *Ordinalzahl*, falls x transitiv ist, und für alle $y, z \in x$ gilt, dass $y \in z \vee y = z \vee z \in y$.

Beispiel. Die leere Menge $\emptyset$ ist trivialerweise eine Ordinalzahl.

Lemma 2.1. *(i) Falls α eine Ordinalzahl ist und $\beta \in \alpha$, dann ist auch β eine Ordinalzahl.*
(ii) Falls $\alpha \neq \beta$ Ordinalzahlen sind und $\alpha \subseteq \beta$, dann ist $\alpha \in \beta$.
(iii) Falls α und β Ordinalzahlen sind, gilt $\alpha \subseteq \beta$ oder $\beta \subseteq \alpha$.

Beweis. (i) ist trivial.
(ii) Betrachte $\beta - \alpha := \{x \in \beta : \neg\, x \in \alpha\} \neq \emptyset$. Diese Menge hat ein $\in$-minimales Element $\gamma \in \beta - \alpha$, sodass für alle $\delta \in \beta - \alpha$ mit $\delta \neq \gamma$ gilt $\gamma \in \delta$. Es gilt $\alpha = \{\xi \in \beta : \xi < \gamma\} = \gamma$.
(iii) Interessant ist nur der Fall $\alpha \neq \beta$. Offensichtlich ist $\gamma := \alpha \cap \beta \neq \emptyset$ wieder eine Ordinalzahl. Es gilt $\gamma = \alpha$ oder $\gamma = \beta$. Tatsächlich, denn sonst würde aus (ii) folgen, dass $\gamma \in \alpha$ und $\gamma \in \beta$. Das würde aber bedeuten $\gamma \in \alpha \cap \beta = \gamma$, was das Fundierungsaxiom aus ZFC verletzt. □

Lemma 2.2. *Für jede Ordinalzahl α ist auch $\alpha + 1 := \alpha \cup \{\alpha\}$ eine Ordinalzahl.*

Beweis. Zunächst ist $\alpha + 1$ transitiv. Denn für ein beliebiges Element $y \in \alpha \cup \{\alpha\}$ kann man unterscheiden: Entweder gilt $y \in \alpha$, dann folgt aus der Transitivität von α, dass $y \subseteq \alpha \subseteq \alpha + 1$. Oder $y = \alpha$, was ebenfalls $y \subseteq \alpha + 1$ impliziert.
Auch die Vergleichbarkeit von zwei beliebigen Elementen $y, z \in \alpha + 1$ ist leicht zu

sehen. Falls $y, z \in \alpha$ gilt, ist sie darauf zurückzuführen, dass α eine Ordinalzahl ist. Falls $y \in \alpha$ und $z = \alpha$, gilt $y \in z$. Der umgekehrte Fall verhält sich analog; und falls $y = \alpha$ und $z = \alpha$, gilt natürlich $y = z$. □

Lemma 2.3. *(i) Jede von Neumann-Zahl $n \in \omega$ ist eine Ordinalzahl.*
(ii) ω ist eine Ordinalzahl.

Beweis. (i) folgt sofort aus dem vorherigen Lemma und der Tatsache, dass $\emptyset$ eine Ordinalzahl ist.
(ii) Die Transitivität kann man induktiv zeigen: Zunächst gilt $\emptyset \subseteq \omega$. Nehmen wir nun an, dass für $n \in \omega$ bereits gilt $n \subseteq \omega$, und sei $x \in n + 1 = n \cup \{n\}$ beliebig. Dann ist entweder $x \in n$, also nach Annahme auch $x \in \omega$; oder $x = n$, also ebenfalls $x \in \omega$. Somit folgt $n+1 \subseteq \omega$. Die Vergleichbarkeit zweier Elemente $n, m \in \omega$ folgt daraus, dass n und m Ordinalzahlen sind, und aus Lemma 2.1 (iii) und (ii). □

Definition. Eine Ordinalzahl $\alpha \neq 0$ heißt *Nachfolgerordinalzahl*, falls es eine Ordinalzahl β gibt, sodass $\alpha = \beta + 1$. Andernfalls heißt α *Limesordinalzahl*.

Beispiel. Alle $n \in \omega$ mit $n \neq 0$ sind Nachfolgerordinalzahlen.
ω ist eine (die kleinste) Limesordinalzahl.
$\omega + 1$ ist eine Nachfolgerordinalzahl.

Definition. Eine zweistellige Relation R auf einer Menge M heißt *lineare Ordnung*, falls gilt:
(i) $\forall x \neg xRx$ (Antireflexivität)
(ii) $\forall x \forall y \forall z\ xRy \wedge yRz \rightarrow xRz$ (Transitivität)
(iii) $\forall x \forall y\ xRy \vee x = y \vee yRx$ (Vergleichbarkeit)

Definition. Eine zweistellige Relation R auf einer Menge M heißt *fundiert*, falls jede nichtleere Teilmenge $A \subseteq M$ ein R-minimales Element $x \in A$ besitzt, dh. für alle $y \in A$ gilt $\neg yRx$.

Definition. Eine *Wohlordnung* ist eine fundierte lineare Ordnung. Wohlordnungen werden üblicherweise mit dem Symbol $<$ bezeichnet.

Lemma 2.4. *(i) Sei α eine Ordinalzahl. Dann ist $\in$ eine Wohlordnung auf α.*
(ii) Sei X eine nichtleere Menge von Ordinalzahlen. Dann ist $\in$ eine Wohlordnung auf X.

Beweis. (i) Transitivität und Vergleichbarkeit sind in der Definition von Ordinalzahl enthalten. Antireflexivität und Fundiertheit folgen aus dem Fundierungsaxiom aus ZFC.
(ii) Transitivität: Seien $x, y, z \in X$ mit $x \in y$ und $y \in z$. Dann ist $x \in z$, da z transitiv ist. Vergleichbarkeit folgt aus Lemma 2.1, und der Rest wieder aus dem Fundierungsaxiom. □

Bemerkung. Ab jetzt werden für Ordinalzahlen $\alpha \in \beta$ und $\alpha < \beta$ austauschbar verwendet.

Definition. Eine Menge A heißt *abzählbar*, wenn es eine injektive Abbildung $f : A \to \omega$ gibt, und *abzählbar unendlich*, wenn es eine bijektive Abbildung $f : A \to \omega$ gibt. Eine Menge heißt *überabzählbar*, wenn sie nicht abzählbar ist.

Beispiel. $\omega + 1$ ist abzählbar, da $f : \omega + 1 \to \omega$ mit $f(\omega) = 0$ und $f(n) = n + 1$ für alle $n \in \omega$ eine bijektive Abbildung ist. (Analog ist $\omega + n$ für alle $n \in \omega$ abzählbar.)
$\omega \cdot 2 := \omega + \omega$ ist abzählbar. Als Bijektion kann man etwa $f : \omega + \omega \to \omega$ mit $f(n) = 2n$ und $f(\omega + n) = 2n + 1$ wählen. (Analog sind alle $\omega \cdot n$ mit $n \in \omega$ abzählbar.)

Definition. Die kleinste Ordinalzahl, die überabzählbar ist, wird mit ω_1 bezeichnet.

Lemma 2.5. *ω_1 ist eine Limesordinalzahl.*

Beweis. Angenommen ω_1 ist ein Nachfolger, das heißt es gibt eine Ordinalzahl α, sodass $\omega_1 = \alpha + 1$. Klarerweise ist α unendlich, aber wegen $\alpha < \alpha + 1$ und der Definition von ω_1 ist α abzählbar. Die Abbildung $f : \alpha + 1 \to \alpha$ mit

$$f(\beta) = \begin{cases} 0, & \text{falls } \beta = \alpha \\ \beta + 1, & \text{falls } \beta \in \omega \\ \beta, & \text{sonst} \end{cases}$$

ist bijektiv. Zusammen mit der Bijektion zwischen α und ω ergibt das, dass auch ω_1 abzählbar ist; das ist ein Widerspruch. □

3 Turingmaschinen

Eine (deterministische) Turingmaschine ist ein Objekt der theoretischen Informatik, das die Arbeitsweise eines Computers modelliert. Es gibt viele äquivalente Definitionen der Turingmaschine, im Folgenden wird eine dreispurige Version beschrieben: Die Maschine hat drei unendlich lange Bänder von Feldern, die als Input-, Arbeits- und Outputband bezeichnet werden. Die Felder enthalten jeweils genau ein Zeichen aus einem vorgegebenen Alphabet, wobei ein „leeres“ Feld oft durch ein spezielles Zeichen wie □ repräsentiert wird. Die Maschine befindet sich zu jedem Zeitpunkt in genau einem von mehreren Zuständen, der in einem Register festgehalten wird. Außerdem gibt es einen (mehrteiligen) Lese- und Schreibkopf (in der Abbildung: „head“), der auf jedem der Bänder zu jedem

Zeitpunkt ein Feld besetzt und das dortige Zeichen einlesen und überschreiben kann. Ebenso kann er den aktuellen Zustand einlesen und verändern.

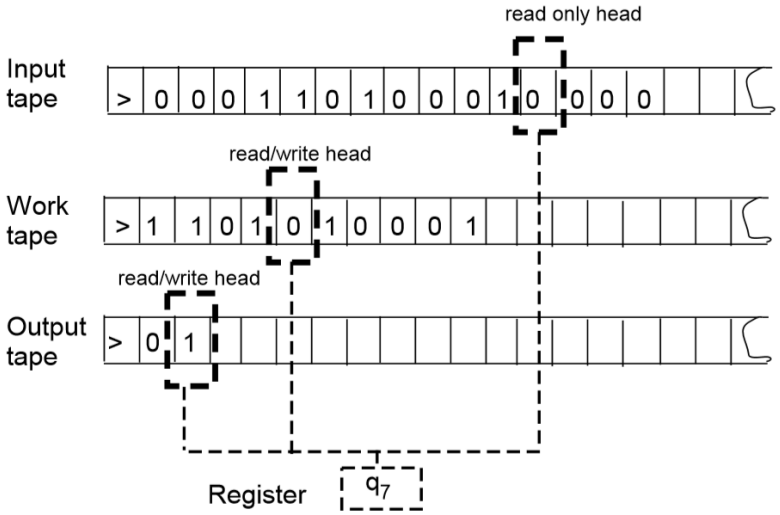

Abbildung 1: Eine 3-spurige Turingmaschine [1, S. 11]

Um ein Programm mit einer bestimmten Eingabe auszuführen, wird das Inputband mit einer endlichen Zeichenkette beschrieben (gefolgt von unendlich vielen Leerzeichen), die anderen beiden Bänder bleiben leer, in dem Register wird ein spezieller Startzustand ausgewählt, und der Lesekopf wird auf allen drei Bändern auf das erste Feld gesetzt. In der Abbildung sind diese Felder für das spezielle Zeichen > reserviert, das die Anweisung zum Starten der Berechnung symbolisieren soll. Das dient aber nur der Anschaulichkeit und kann bei der Definition auch weggelassen werden.
Die Maschine folgt dann den Anweisungen einer Überführungsfunktion, die dem konkreten Computerprogramm entspricht, und ändert schrittweise die Zeichen auf den Bändern. Ein Schritt besteht daraus, den aktuellen Zustand sowie die drei besetzten Felder einzulesen und daraufhin die dortigen Zeichen zu überschreiben (oder unverändert zu lassen), einen neuen (oder den gleichen) Zustand einzustellen, und den Lesekopf auf jedem Band um ein Feld nach links oder rechts zu bewegen oder die Position gleich zu lassen. Es ist naheliegend, die Schritte mithilfe der natürlichen Zahlen zu nummerieren, und tatsächlich kann die Turingmaschine „beliebig lange“ laufen, in dem Sinn, dass eine beliebig große endliche Anzahl von Schritten durchgeführt wird. Falls in irgendeinem Schritt der Endzustand eingelesen wird, hält die Turingmaschine an und das Programm ist vollständig ausgeführt. Das Ergebnis kann dann am Outputband abgelesen werden.

Definition. Sei $k \geq 2$. Eine *k-spurige Turingmaschine* ist ein Tripel $\mathcal{T} = (A, Q, \delta)$, bestehend aus:
einem *Alphabet* $A = \{0, 1, \square\}$;
einer endlichen *Zustandsmenge* Q, die die beiden speziellen Zustände q_A und q_E (*Anfangs- bzw. Endzustand*) enthält;
und einer *Überführungsfunktion* $\delta : (Q \backslash \{q_E\}) \times A^k \to Q \times A^{k-1} \times \{L, S, R\}^k$.
Die Menge der möglichen Inputs für $\mathcal{T}$ ist $A^* = \bigcup_{n \in \mathbb{N}} A^n$, also die Menge aller endlichen Folgen von Zeichen aus A.

Bemerkung. Im Zielbereich der Überführungsfunktion ist der Exponent von A nur $k-1$, da die Zeichen auf dem Inputband zwar gelesen, aber nicht verändert werden sollen. (Bei einer einspurigen Turingmaschine wird das Inputband natürlich schon überschrieben, die Definition ist entsprechend abzuwandeln.) Ein Element von $\{L, S, R\}^k$ repräsentiert, ob der Lesekopf auf jedem der k Bänder nach links (L) oder rechts (R) bewegt oder nicht verändert (S, für „stay“) wird.

Bemerkung. Wie bereits erwähnt gibt es viele andere Definitionen der Turingmaschine, die in dem Sinn äquivalent sind, dass sie genau die gleichen Berechnungen durchführen können (aber nicht unbedingt gleich schnell). So kann man zum Beispiel auf einer einspurigen Turingmaschine leicht die Arbeitsweise einer dreispurigen simulieren, indem man jeweils jedes dritte Feld als Teil des Input-, Arbeits- und Outputbandes behandelt. Außerdem kann man jedes beliebige endliche Alphabet einfach mit $\{0, 1, \square\}$ kodieren, und es würde auch ausreichen, nur $\{0, 1\}$ zu verwenden.

Definition. Als *Konfiguration* wird die vollständige Beschreibung einer Turingmaschine zu einem bestimmten Zeitpunkt in der Berechnung bezeichnet. Sie gibt den aktuellen Zustand, die Position des Lesekopfes, und den ganzen Inhalt aller Bänder (nicht nur der Felder unter dem Lesekopf) an.

Definition. Man sagt, eine Turingmaschine $\mathcal{T}$ *terminiert* oder *hält (nach endlich vielen Schritten)* für die Eingabe $w \in A^*$, falls es ein $n \in \omega$ gibt, sodass im n-ten Schritt der Endzustand q_E eingelesen wird.

Es ist einfach zu sehen, dass nicht jedes Programm mit jedem Input hält:

Beispiel. Sei δ eine Überführungsfunktion einer einspurigen Turingmaschine, die die Anweisungen $(q_A, 1) \mapsto (q_2, 1, R)$, $(q_1, 1) \mapsto (q_2, 1, R)$ und $(q_2, 1) \mapsto (q_1, 1, L)$ enthält. Führt man dieses Programm mit dem Input $w = 11 = (1, 1)$ aus, dann wird im nullten Schritt das erste Feld eingelesen, nicht verändert, der Lesekopf nach rechts bewegt, und der Zustand q_2 gewählt. Im nächsten Schritt wird wieder kein Zeichen verändert und der Kopf zurück nach links bewegt, usw. Die Maschine hält mit diesem Input also nicht.

Definition. Eine n-stellige Funktion $f : D \subseteq (A^*)^n \to A^*$ $(n \geq 1)$ heißt *(Turing-)berechenbar*, falls es eine Turingmaschine $\mathcal{T}$ gibt, die Folgendes erfüllt: Wandelt man ein beliebiges Element $(x_1, ..., x_n) \in D$ zu einer Eingabe $x_1 \square x_2 \square ... \square x_n \square \square ...$ um, dann hält die Maschine mit dieser Eingabe nach endlich vielen Schritten im Endzustand, und auf dem Outputband steht der Funktionswert $f(x_1, ..., x_n)$ gefolgt von Leerzeichen. Mit jedem Input, für den f nicht definiert ist, terminiert die Maschine nicht.

Beispiel (Unäre Addition). Für eine natürliche Zahl n sei $[n]_1$ die Unärdarstellung von n als Zeichenkette von $n + 1$ Einsen. Dann ist die Additionsfunktion $f : \{1\}^* \times \{1\}^* \to \{1\}^*$ mit $f([n]_1, [m]_1) = [n + m]_1$ berechenbar. Dazu sei eine einspurige Turingmaschine mit Zustandsmenge $\{q_1, q_2, q_3, q_A, q_E\}$ gegeben, die das Leerzeichen zwischen den beiden eingegebenen Zahlen durch eine Eins ersetzt, und die letzten beiden Einsen löscht. Das kann zum Beispiel durch folgende Überführungsfunktion beschrieben werden:

$$\begin{array}{lll} \delta : (q_A, 1) & \mapsto (q_A, 1, R) & \qquad (q_1, \square) \mapsto (q_2, \square, L) \\ (q_A, \square) & \mapsto (q_1, 1, R) & \qquad (q_2, 1) \mapsto (q_3, \square, L) \\ (q_1, 1) & \mapsto (q_1, 1, R) & \qquad (q_3, 1) \mapsto (q_E, \square, S) \end{array}$$

Bemerkung. Man kann schnell sehen, dass es Funktionen geben muss, die nicht berechenbar sind. Mehr noch, unter allen möglichen Funktionen ist der Anteil der berechenbaren Funktionen sogar sehr gering. Denn es gibt nur abzählbar viele verschiedene Turingmaschinen, also auch nur abzählbar viele verschiedene berechenbare Funktionen. Andererseits ist schon die Menge aller Funktionen $f : \omega \to \{0, 1\}$ bekanntermaßen gleichmächtig wie die Menge der reellen Zahlen, also überabzählbar.

Definition. Eine Menge $M \subseteq A^*$ heißt *entscheidbar*, wenn ihre charakteristische Funktion $\chi_M : A^* \to \{0, 1\}$ definiert durch

$$\chi_M(x) = \begin{cases} 1, & \text{falls } x \in M \\ 0, & \text{sonst} \end{cases}$$

berechenbar ist.

Bemerkung. Dass eine Menge entscheidbar ist, bedeutet also, dass es eine Turingmaschine gibt, die für jeden Input in endlich vielen Schritten entscheidet, ob er in der Menge enthalten ist oder nicht.

Mit diesen grundlegenden Begriffen lässt sich bereits eine interessante Überlegung formulieren, nämlich das Halteproblem: Wie bereits beobachtet, gibt es Paare von Turingmaschinen und Eingaben, die nach endlich vielen Schritten den

Endzustand erreichen, und solche die das nicht tun. Die Frage ist nun, ob es eine Möglichkeit gibt, für ein beliebiges solches Paar zu bestimmmen, welcher der beiden Fälle eintreten wird. Eine Turingmaschine wird eindeutig beschrieben durch ihre endliche Zustandsmenge und ihre Überführungsfunktion, wobei man letztere ebenfalls als endliche Menge betrachten kann. Daher kann man durch eine geeignete Kodierung jedem Paar von Turingmaschine und Eingabe eine Zeichenkette in A^* zuordnen. Damit lässt sich das Halteproblem wie folgt formal formulieren: Ist die Menge aller Paare von Turingmaschine und Input, die nach endlich vielen Schritten halten, entscheidbar?

Satz 3.1 (Halteproblem, Turing 1936)**.** *Es gibt keine Turingmaschine $\mathcal{H}$, die für eine beliebige Turingmaschine $\mathcal{T}$ und eine beliebige Eingabe w entscheidet, ob $\mathcal{T}$ mit w nach endlich vielen Schritten anhält oder nicht.*

Beweis. Angenommen, es gibt so eine Turingmaschine $\mathcal{H}$, die mit jeder Eingabe der Form $(\mathcal{T}, w)$ anhält, mit der Ausgabe:

1, falls $\mathcal{T}$ mit w hält;

0, falls $\mathcal{T}$ mit w nicht hält.

Insbesondere funktioniert $\mathcal{H}$ auch für Eingaben der Form $(\mathcal{T}, \mathcal{T})$, also für eine Turingmaschine, die mit sich selbst als Input läuft. Darauf aufbauend kann man eine weitere Turingmaschine $\mathcal{H}'$ definieren, die sich mit einer Turingmaschine $\mathcal{T}$ als Eingabe folgendermaßen verhält:

sie hält nicht, falls $\mathcal{T}$ mit Eingabe $\mathcal{T}$ hält;

sie hält (z.B. mit Ausgabe 1), falls $\mathcal{T}$ mit Eingabe $\mathcal{T}$ nicht hält.

Nun zeigt sich ein Problem, wenn man die Turingmaschine $\mathcal{H}'$ mit der Eingabe $\mathcal{H}'$ betrachtet. Falls die Berechnung hält, dann hält sie nach der Definition von $\mathcal{H}'$ nicht. Andererseits, falls die Berechnung nicht hält, dann müsste sie doch halten. Es gibt also keine Möglichkeit, die nicht zu einem Widerspruch führt, daher kann es $\mathcal{H}'$ und folglich auch $\mathcal{H}$ nicht geben. □

4 Ordinalzahl-Turingmaschinen

Das Konzept von Turingmaschinen kann erweitert werden, um Maschinen zu beschreiben, die unendlich viele Rechenschritte durchführen können. Dazu wird der Begriff der Ordinalzahl aus Abschnitt 2 verwendet: So wie man die einzelnen Schritte einer gewöhnlichen Turingmaschine durch die natürlichen Zahlen beschreiben kann, soll bei diesem Modell jeder Ordinalzahl ein Rechenschritt entsprechen.

4.1 Funktionsweise und Entscheidbarkeit

Eie Ordinalzahl-Turingmaschine hat die gleiche „Hardware“ wie eine übliche 3-spurige Turingmaschine, also Input-, Arbeits- und Outputband, ein Register für den aktuellen Zustand, und einen Lese- und Schreibkopf. Die Maschine folgt einem endlichen Algorithmus und hat nur endlich viele verschiedene Zustände. Wesentlich ist das Verhalten der Ordinalzahl-Turingmaschine in Limesschritten, also Schritten, die einer Limesordinalzahl entsprechen: Dort tritt ein spezieller Limeszustand ein, und für jedes Feld wird ein Grenzwert aus den Werten gebildet, die das Feld bis zu diesem Zeitpunkt hatte.

Definition. Eine *Ordinalzahl-Turingmaschine* ist ein Tripel $\mathcal{T}_O = (A, Q, \delta)$, bestehend aus:
einem *Alphabet* $A = \{0, 1\}$;
einer endlichen *Zustandsmenge* Q, die zusätzlich zu q_A und q_E einen *Limeszustand* q_L enthält;
und einer *Überführungsfunktion* $\delta : (Q \backslash \{q_E\}) \times A^3 \to Q \times A^2 \times \{L, S, R\}^3$, bestehend aus endlich vielen Anweisungen.
Als Eingabe ist jede unendliche Zeichenkette aus 0 und 1 zugelassen.
Falls $\alpha = 0$ oder α eine Nachfolgerordinalzahl ist, besteht der Schritt α daraus, die Felder unter dem Lesekopf, den Zustand im Register und die Postition des Lesekopfes gemäß der Überführungsfunktion δ zu verändern.
Falls α eine Limesordinalzahl ist, tritt der Zustand q_L ein und der Lesekopf wird auf allen Bändern ganz nach links gesetzt. Nun wird für jedes Feld der Limes superior der Werte gebildet, die das Feld in den Schritten $< \alpha$ hatte. Das heißt falls die Werte irgendwann vor dem Limesschritt nur noch 0 oder nur noch 1 sind, behält das Feld diesen Wert. Ansonsten, falls sie bis zum Schritt α unbeschränkt oft zwischen 0 und 1 wechseln, wird im Limesschritt 1 eingetragen. Der Prozess wird für alle Felder wiederholt, ohne den Lesekopf zu bewegen.
Die Maschine *hält* mit einem bestimmten Input, falls in irgendeinem Schritt der Endzustand q_E eintritt.

Bemerkung. Die möglichen Eingaben für eine Ordinalzahl-Turingmaschine sind genau alle unendlichen Binär-Folgen $x \in 2^\omega$. Da eine bijektive Entsprechung zwischen 2^ω und $\mathbb{R}$ besteht, kann man jede Eingabe als Binärdarstellung einer reellen Zahl interpretieren.

Die folgenden Begriffe sind analog zu denen für gewöhnliche Turingmaschinen definiert.

Definition. Sei $n \geq 1$. Eine n-stellige Funktion $f : \mathbb{R}^n \to \mathbb{R}$ heißt *in unendlicher Zeit berechenbar*, falls es eine Ordinalzahl-Turingmaschine $\mathcal{T}_O$ gibt, für die Folgendes gilt:

Seien $x_1, ..., x_n \in \mathbb{R}$ beliebig, und für $i \in \{1, ..., n\}$ sei $x_i = x_{i,1}x_{i,2}x_{i,3}...$ die Binärdarstellung von x_i. Dann erreicht $\mathcal{T}_O$ mit dem Input $x_{1,1}...x_{n,1}x_{1,2}...x_{n,2}x_{1,3}...$ nach irgendeiner Ordinalzahl von Schritten den Endzustand, und auf dem Ausgabeband steht die Binärdarstellung von $f(x_1, x_2, ..., x_n)$.

Definition. Eine Teilmenge M der reellen Zahlen heißt *in unendlicher Zeit entscheidbar*, falls ihre charakteristische Funktion $\chi_M : \mathbb{R} \to \{0, 1\}$ in unendlicher Zeit berechenbar ist.

Satz 4.1. *Falls eine Ordinalzahl-Turingmaschine mit einer Eingabe hält, dann tritt der Endzustand bereits nach abzählbar vielen Schritten ein.*

Beweis. Angenommen eine Berechnung einer Ordinalzahl-Turingmaschine läuft seit überabzählbar vielen Schritten. Man betrachte die Konfiguration nach dem Schritt ω_1. Wie in jedem Limesschritt befindet sich die Maschine im Zustand q_L und der Lesekopf ganz links. Für jedes Feld, das nach Schritt ω_1 den Wert 0 hat, muss es einen früheren (also abzählbaren) Zeitpunkt geben, ab dem dieses Feld bis ω_1 immer den Wert 0 hat. Wenn ein Feld hingegen nach Schritt ω_1 den Wert 1 hat, gibt es zwei Möglichkeiten: Entweder das Feld hat seit irgendeinem früheren Zeitpunkt unverändert den Wert 1, oder aber der Wert alterniert bis zum Schritt ω_1 unbeschränkt oft. Da es nur abzählbar viele Felder gibt, kann man all jene Felder, die bis zum Zeitpunkt ω_1 konstant werden, nummerieren als F_n für $n \in \omega$. Dann gibt es für jedes $n \in \omega$ ein $\beta_n \in \omega_1$ (abzählbar), sodass F_n schon ab dem Schritt β_n konstant ist. Sei jetzt $\alpha_0 = sup\{\beta_n : n \in \omega\} = \bigcup\{\beta_n : n \in \omega\}$. Das ist eine Vereinigung von abzählbar vielen abzählbaren Mengen, also abzählbar. Alle Felder, die bis ω_1 konstant werden, sind es bereits bei α_0.
Nach α_0 ändern sich also nur noch die Felder, die bis ω_1 unbeschränkt oft alternieren. Es gibt eine abzählbare Ordinalzahl α_1 (mit $\alpha_0 < \alpha_1$), sodass alle diese Felder zwischen α_0 und α_1 mindestens einmal ihren Wert ändern. Rekursiv kann man eine Folge $\alpha_1 < \alpha_2 < ...$ von abzählbaren Ordinalzahlen finden, sodass für alle $n \in \omega$ jedes unbeschränkt oft alternierende Feld zwischen α_n und α_{n+1} mindestens einmal den Wert ändert. Sei nun $\delta = sup\{\alpha_n : n \in \omega\}$. Das ist eine abzählbare Limesordinalzahl, also ist wie bei ω_1 der Lesekopf ganz links und der Zustand q_L. Alle Felder, die vor ω_1 konstant werden, werden bereits vor δ konstant, und alle die bis ω_1 unbeschränkt oft alternieren, tun das auch bis δ - das heißt ein Feld enthält genau dann eine 1 im Schritt δ, wenn sie im Schritt ω_1 eine 1 enthält. Somit stimmen die Konfigurationen vollständig überein, und die Berechnung wiederholt sich. Darüber hinaus kann ein Feld, das zum Zeitpunkt δ den Wert 0 hat, zwischen δ und ω_1 nicht mehr den Wert 1 annehmen. Dadurch ist sichergestellt, dass die Berechnung auch durch Limesbildung nie aus der Schleife entkommt. □

Bemerkung. Im letzten Teil des Beweises genügt es nicht zu beobachten, dass zweimal die gleiche Konfiguration auftritt. Im Gegensatz zu gewöhnlichen Turingmaschinen lässt sich daraus noch nicht folgern, dass die Berechnung für immer in einer Schleife gefangen ist und somit nicht hält. Zum Beispiel kann man sich eine Berechnung vorstellen, bei der in den ersten ω Schritten nur ein Feld immer zwischen 0 und 1 alterniert und sich der Zustand nicht ändert. Aber im Limesschritt ω tritt ein neuer Zusand (der Limeszustand) ein, und dadurch ist es möglich, dass die Berechnung die Schleife verlässt, und sogar irgendwann hält.
Für eine nie entkommbare Schleife ist es also notwendig, dass die gleiche Konfiguation bei zwei *Limes*ordinalzahlen $\alpha < \beta$ auftritt. Außerdem muss auch der Limes der sich wiederholenden Konfigurationen wieder dieselbe Konfiguration sein. Das ist sicher der Fall, wenn alle Felder, die im Schritt α eine 0 enthalten, bis zum Schritt β nicht mehr den Wert ändern.

Eine erste Beobachtung über die Leistungsfähigkeit von Ordinalzahl-Turingmaschinen: Das Halteproblem für gewöhnliche Turingmaschinen ist mit Ordinalzahl-Turingmaschinen lösbar.

Satz 4.2. *Die Menge der Paare $(\mathcal{T}, w)$ aus einer Turingmaschine und einem Input, die nach endlich vielen Schritten halten, ist in unendlicher Zeit entscheidbar.*

Beweis. Sei $\mathcal{H}_O$ eine Ordinalzahl-Turingmaschine, die folgendermaßen definiert ist: In den ersten ω Schritten simuliert sie die Rechenschritte von $(\mathcal{T}, w)$. Der Endzustand von $\mathcal{T}$ wird dabei in der Zustandsmenge von $\mathcal{H}_O$ durch $\tilde{q}_E$ repräsentiert. Falls der Zustand $\tilde{q}_E$ eintritt, soll 1 auf das Outputband von $\mathcal{H}_O$ geschrieben und $\mathcal{H}_O$ angehalten (Zustand q_E) werden. Sonst, falls $\tilde{q}_E$ in keinem Schritt $n \in \omega$ eintritt (also $\mathcal{T}$ nicht hält), erreicht $\mathcal{H}_O$ ohne Halten den Schritt ω. Man kann also definieren, dass im Schritt $\omega + 1$ immer 0 auf das Outputband geschrieben und der Zustand q_E angenommen wird. Mit diesen Eigenschaften ist $\mathcal{H}_O$ eine Ordinalzahl-Turingmaschine, die die gesuchte charakteristische Funktion in unendlicher Zeit berechnet. □

Bemerkung. Da ω ein Limesschritt ist, ist das Verhalten von Ordinalzahl-Turingmaschinen dort bereits definiert. Daher kann man $\mathcal{H}_O$ erst im Schritt $\omega + 1$ halten.

Definition. Eine *arithmetische Formel* ist eine prädikatenlogische Formel in der Sprache der Peano Arithmetik.

In der Sprache der Peano Arithmetik gibt es zwei beschränkte Quantoren $\exists x < t$ und $\forall x < t$, die folgendermaßen definiert sind:

$\exists x < t\ \varphi :\Leftrightarrow \exists x(x < t \wedge \varphi)$ und
$\forall x < t\ \varphi :\Leftrightarrow \forall x(x < t \rightarrow \varphi)$, wobei φ eine Formel ist und t ein Term, in dem x nicht vorkommt.

Definition (Arithmetische Hierarchie)**.** Sei φ eine arithmetische Formel.
Falls φ logisch äquivalent ist zu einer Formel, in der alle vorkommenden Quantoren beschränkt sind, dann ist φ in Σ_0^0 und Π_0^0.
Falls es eine natürliche Zahl k, Variablen $m_1, ..., m_k$ und eine Formel $\psi \in \Pi_n^0$ gibt, sodass φ logisch äquivalent ist zu $\exists m_1 \exists m_2 ... \exists m_k \psi$, dann ist φ in Σ_{n+1}^0.
Falls es eine natürliche Zahl k, Variablen $m_1, ..., m_k$ und eine Formel $\psi \in \Sigma_n^0$ gibt, sodass φ logisch äquivalent ist zu $\forall m_1 \forall m_2 ... \forall m_k \psi$, dann ist φ in Π_{n+1}^0.

Bemerkung. Falls $\varphi \in \Sigma_n^0$ oder $\varphi \in \Pi_n^0$ für eine natürliche Zahl n, dann gilt auch $\varphi \in \Sigma_m^0$ und $\varphi \in \Pi_m^0$ für alle $m > n$.
Jede arithmetische Formel ist in der Klasse Σ_n^0 für mindestens eine natürliche Zahl n.

Definition. Eine Teilmenge X der natürlichen Zahlen wird durch die Formel $\varphi(x)$ *definiert*, falls für alle natürlichen Zahlen n gilt: $n \in X$ genau dann wenn $\mathfrak{N} \models \varphi(n)$, wobei $\mathfrak{N}$ das Standardmodell der natürlichen Zahlen ist.

Definition. Eine Menge X von natürlichen Zahlen heißt Σ_n^0 (bzw. Π_n^0), falls sie durch eine Σ_n^0-Formel (bzw. Π_n^0-Formel) definiert werden kann. X heißt *arithmetische Menge*, falls sie durch eine arithmetische Formel definiert werden kann.

Satz 4.3. *Die Menge der arithmetischen Formeln, die wahr sind, ist in unendlicher Zeit entscheidbar.*

Beweis. Das ist durch eine Induktion über den Formelaufbau einfach zu sehen. Es ist bekannt, dass mit einer gewöhnlichen Turingmaschine für zwei natürliche Zahlen n und m die Addition, Multiplikation und Exponentiation durchgeführt, sowie die Relation $n < m$ überprüft werden können. Ist weiters der Wahrheitswert der Formeln φ und ψ bekannt, kann sicher auch der von $\neg\varphi$ und $\varphi \wedge \psi$ ermittelt werden. Schließlich, für eine Formel der Form $\exists n\ \varphi(n, \vec{x})$ kann eine Ordinalzahl-Turingmaschine die ω vielen möglichen Werte für n durchprobieren und jeweils den Wahrheitswert von $\varphi(n, \vec{x})$ überprüfen. □

Korollar 4.4. *Jede arithmetische Menge ist in unendlicher Zeit entscheidbar.*

Lemma 4.5. *Man kann die zweistelligen Relationen auf ω als reelle Zahlen kodieren.*

Beweis. Sei $R \subseteq \omega \times \omega$ eine Relation. Sei $\pi : \omega \times \omega \rightarrow \omega$ eine bijektive Abbildung (z.B. die Cantorsche Paarungsfunktion). Dann kann man R den unendlichen

Binärstring $x = x_1x_2x_3...$ zuordnen, wobei $x_{\pi(n,m)} = 1$ genau dann, wenn nRm. Auf diese Art wird jeder zweistelligen Relation genau eine reelle Zahl zugeordnet. □

Definition. Mit WO $\subseteq \mathbb{R}$ wird die Menge der reellen Zahlen bezeichnet, die eine Wohlordnung auf ω kodieren.

Satz 4.6. WO *ist in unendlicher Zeit entscheidbar.*

Beweis. Gesucht ist eine Ordinalzahl-Turingmaschine, die für eine beliebige reelle Zahl x als Binärstring auf dem Inputband entscheidet, ob die dadurch kodierte Relation R_x eine Wohlordnung ist. Zunächst wird überprüft, ob es sich um eine lineare Ordnung handelt.
Antireflexivität: Dazu wird einfach das Inputband einmal durchlaufen, und falls eine 1 an einer Stelle auftritt, die nR_xn für ein $n \in \omega$ repräsentiert, wird die Berechnung mit dem Output 0 angehalten. Falls andererseits der erste Limesschritt ohne Halten erreicht wird, ist klar, dass die Relation antireflexiv ist.
Vergleichbarkeit: Das Inputband wird noch einmal durchlaufen. Immer wenn eine 0 auftritt, d.h. $\neg\, nR_xm$ für gewisse $n, m \in \omega$, wird zunächst überprüft, ob $n = m$ gilt. Falls nicht, wird der Lesekopf zu der Stelle bewegt, die (m, n) repräsentiert. Falls dort auch eine 0 steht, handelt es sich nicht um eine lineare Ordnung und die Berechnung hält. Sonst wird der Lesekopf zurück zu (n, m) bewegt und der Input weiter durchlaufen. Der Test ist wieder abgeschlossen, wenn der Limes erreicht wird.
Transitivität: Um auch diesen Test in nur ω vielen Schritten durchzuführen, werden alle Paare von Feldern des Inputbandes durchlaufen. Relevant sind dabei nur Paare, die Relationen nR_xm und mR_xk repräsentieren, und wo beide Felder 1 enthalten. In solchen Fällen muss überprüft werden, ob auch nR_xk gilt.

Falls x diese drei Tests bestanden hat, muss noch überprüft werden, ob die Ordnung fundiert ist. Zunächst wird in ω vielen Schritten versucht, ein R_x-minimales Element zu finden. Dazu wird der Input durchlaufen, und die aktuelle Vermutung n auf dem Arbeitsband festgehalten. Jedes Mal wenn eine Zahl gefunden wird, die ein Vorgänger von n bezüglich R_x ist, wird die Vermutung durch diese Zahl ersetzt. Außerdem wird ein spezielles Markierungs-Feld auf dem Arbeitsband jedes Mal alterniert („ein-/ausgeschaltet“), wenn die Vermutung geändert wird. Falls im Limesschritt dieses Feld eine 1 enthält, also die Vermutung unbeschränkt oft geändert wurde, gibt es kein minimales Element, also ist R_x keine Wohlordnung. Falls das Feld eine 0 enthält, befindet sich eine Zahl n auf dem Arbeitsband, die das korrekte minimale Element darstellt. Damit ist sichergestellt, dass alle Teilmengen von ω, die n enthalten, ein R_x-minimales

Element haben. Daher wird in den nächsten ω Schritten jedes Feld der Eingabe gelöscht (Null gesetzt), das eine Relation mit n beschreibt, da dort nichts mehr zu überprüfen ist. (Streng genommen kann das Eingabeband nicht überschrieben werden, aber zur Übersicht wird hier darauf verzichtet, ein weiteres Arbeitsband zu simulieren, auf dem sich eine Kopie der Eingabe befindet, da das schon in Abschnitt 3 besprochen wurde.)
Anschließend wird von dieser neuen Ordnung wieder das minimale Element gesucht und der Vorgang wird iteriert. Vorsicht: Es ist nicht klar, wie oft der Vorgang iteriert werden muss, bis alle natürlichen Zahlen gelöscht sind. (Bei der üblichen Ordnung durch $\in$ ist es ω mal.) Möglicherweise wird sogar ein Schritt erreicht, der ein Limes von Limesordinalzahlen ist. In so einem Schritt befindet sich keine neue Vermutung auf dem Arbeitsband, sondern ein unsinniger Limes aus den bisher entfernten minimalen Elementen, der gelöscht werden sollte. Um zu erkennen, wann das der Fall ist, wird ein zweites Markierungs-Feld auf dem Arbeitsband bestimmt, das nach jedem Limesschritt ein- und gleich wieder ausgeschaltet wird. Wenn dieses Feld in einem Limesschritt 1 ist, weiß man, dass es sich um einen Limes von Limiten handeln muss. In diesem Fall wird in ω Schritten das gesamte Arbeitsband mit 0 überschrieben, bevor die Iteration normal weitergeht.
Falls irgendwann kein neues minimales Element mehr gefunden werden kann, da die Vermutungen in diesem Schritt nicht konvergieren, handelt es sich nicht um eine Wohlordung. Sonst läuft die Berechnung so lange, bis alle Felder der Eingabe gelöscht sind. Wenn das der Fall ist, also wenn das Eingabeband leer ist, wurde systematisch überprüft, dass jede Teilmenge von ω ein R_x-minimales Element hat. Dann ist R_x eine Wohlordnung, und die Berechnung kann mit Output 1 gehalten werden. □

4.2 Deskriptive Mengenlehre

Im Zusammenhang mit Ordinalzahl-Turingmaschinen ist es interessant, Mengen zu betrachten, die über die arithmetische Hierarchie hinausgehen. Dazu sind einige Grundlagen der deskriptiven Mengenlehre notwendig, die man z.B. in [6], [5, Kapitel 12 und 13] oder [8, Kapitel 7] findet. Der Großteil der Notation sowie der Beweis von Satz 4.8 sind aus [8] übernommen.

In den folgenden Definitionen wird mit ${}^{<\omega}\omega$ die Menge aller endlichen Folgen von natürlichen Zahlen bezeichnet, und mit ${}^{\omega}\omega$ die aller unendlichen. Es ist zu bemerken, dass man die Elemente von ${}^{\omega}\omega$ mit reellen Zahlen bzw. mit unendlichen Binärstrings identifizieren kann.

Definition. Man sagt, eine Folge $s = \langle x_0, x_1, ..., x_{n-1}\rangle \in {}^{<\omega}\omega$ hat die *Länge* $lh(s) := dom(s) = n$, und für $0 \leq m < n$ ist $s \restriction m := \langle x_0, x_1, ..., x_{m-1}\rangle$ die

Einschränkung von s auf Länge m. Die Einschränkung auf Länge 0 ergibt die leere Folge.

Definition. Eine Menge $T \subseteq {}^{<\omega}\omega \times {}^{<\omega}\omega$ heißt *(2-dimensionaler) Baum*, falls:
(i) Für alle $(s,t) \in T$ gilt $lh(s) = lh(t)$, und
(ii) T ist abgeschlossen unter Anfangsstücken, dh. falls $(s,t) \in T$ und $n \leq lh(s)$, dann gilt $(s \restriction n, t \restriction n) \in T$.

Definition. Sei T ein Baum.
(i) Mit $[T]$ wird die Menge aller $(x,y) \in {}^{\omega}\omega \times {}^{\omega}\omega$ bezeichnet, sodass $(x \restriction n, y \restriction n) \in T$ für alle $n \in \omega$.
(ii) Die *Projektion* $p[T]$ ist die Menge aller $x \in {}^{\omega}\omega$, für die es ein $y \in {}^{\omega}\omega$ gibt, sodass für alle $n \in \omega$ gilt $(x \restriction n, y \restriction n) \in {}^{\omega}\omega$.
(iii) Für $x \in {}^{\omega}\omega$ sei T_x die Menge aller $t \in {}^{<\omega}\omega$, sodass $(x \restriction lh(t), t) \in T$.

Bemerkung. Offensichtlich ist $x \in p[T]$ äquivalent zu $\exists y\ (x,y) \in [T]$. Das ist weiter äquivalent dazu, dass T_x beliebig lange Folgen enthält, und dazu, dass $(T_x, \supset)$ nicht fundiert ist.

Definition. Eine Menge $A \subseteq {}^{\omega}\omega$ heißt *analytisch* oder $\mathbf{\Sigma}^1_1$-Menge, falls es einen Baum $T \subseteq {}^{<\omega}\omega \times {}^{<\omega}\omega$ gibt mit $A = p[T]$.
Eine Menge $A \subseteq {}^{\omega}\omega$ heißt *koanalytisch* oder $\mathbf{\Pi}^1_1$-Menge, falls ihr Komplement ${}^{\omega}\omega - A$ analytisch ist.

Satz 4.7. WO *ist eine* $\mathbf{\Pi}^1_1$*-Menge.*

Beweis. Sei LO die Menge aller linearen Ordnungen auf ω und WF die aller fundierten Ordnungen. Dann ist $\mathrm{WO} = \mathrm{LO} \cap \mathrm{WF}$.
Zunächst ist WF eine $\mathbf{\Pi}^1_1$-Menge. Sei dazu der Baum T so definiert:

$$T = \{(s,t) \in {}^{<\omega}\omega \times {}^{<\omega}\omega : [lh(s) = lh(t)] \wedge \\ [\forall i\ (i < lh(s) - 1\ \wedge\ \pi(t_{i+1}, t_i) < lh(s) \rightarrow s_{\pi(t_{i+1},t_i)} = 1)]\}$$

Hier bezeichnet π wieder die Cantorsche Paarungsfunktion. Die Gleichung $s_{\pi(t_{i+1},t_i)} = 1$ ist folgendermaßen zu interpretieren: Falls $x \in {}^{\omega}\omega$ eine reelle Zahl ist, von der s ein Anfangsstück ist, dann erfüllt die von x kodierte Relation $t_{i+1} R_x t_i$.
Es ist einfach zu sehen, dass $p[T] = \mathrm{WF}$ ist. Falls $x \notin \mathrm{WF}$, gibt es eine Folge von natürlichen Zahlen $\langle t_i : i \in \omega \rangle$ mit $t_{i+1} R_x t_i$ für alle $i \in \omega$. Klarerweise ist dann $(T_x, \supset)$ nicht fundiert. Die Umkehrung folgt ebenso.
Da LO durch eine Π^0_1-Formel definiert wird, kann man T schließlich leicht zu einem Baum T' einschränken, der die Projektion $p[T'] = \mathrm{WO}$ hat. $\square$

Satz 4.8. *Sei $A \subseteq {}^{\omega}\omega$ eine beliebige $\mathbf{\Pi}^1_1$-Menge. Dann gibt es eine stetige Funktion $f : {}^{\omega}\omega \to {}^{\omega}\omega$, sodass für alle $x \in {}^{\omega}\omega$ gilt: $x \in A$ gdw $f(x) \in$ WO. Man sagt,* WO *ist $\mathbf{\Pi}^1_1$-vollständig.*

Beweis. Sei $A \subseteq {}^{\omega}\omega$ eine $\mathbf{\Pi}^1_1$-Menge und T der zugehörige Baum mit $p[T] = {}^{\omega}\omega - A$. Dann gilt $x \in A$ genau dann, wenn $(T_x, \supset)$ fundiert ist. Sei $e : \omega \to {}^{<\omega}\omega$ eine bijektive Abbildung, sodass für alle $s \in {}^{<\omega}\omega$ und $i \in \omega$ gilt $e^{-1}(s \restriction i) \leq e^{-1}(s)$. Damit kann man eine von T_x induzierte Ordnung R^x auf ω angeben. (Vorsicht: R^x ist nicht zu verwechseln mit der von x kodierten Ordnung R_x.)

$$\begin{aligned} nR^x m \text{ gdw } [&(e(n) \in T_x \wedge e(m) \in T_x \wedge e(n) \supset e(m)) \vee \\ &(e(n) \in T_x \wedge e(m) \in T_x \wedge e(n) \perp e(m) \wedge e(n) <_{lex} e(m)) \vee \\ &(e(n) \notin T_x \wedge e(m) \in T_x) \vee \\ &(e(n) \notin T_x \wedge e(m) \notin T_x) \wedge n < m)] \end{aligned}$$

Hier ist $s \perp t$ eine Notation für $s \restriction (lh(s) \cap lh(t)) \neq t \restriction (lh(s) \cap lh(t))$, und $<_{lex}$ die lexikographische Ordnung.
Bezüglich R^x sind also Folgen außerhalb von T_x kleiner als solche in T_x, und außerhalb von T_x gilt die übliche Ordnung $<$. Auf T_x entspricht R^x der Kleene-Brouwer-Ordnung, die man als Variante der lexikographischen Ordnung betrachten kann, wobei ein nicht definiertes Folgenglied größer ist als jeder mögliche Wert der Folge. Man kann einfach überprüfen, dass R^x immer eine lineare Ordnung ist.
Sei nun $f : {}^{\omega}\omega \to {}^{\omega}\omega$ so, dass $R_{f(x)} = R^x$, das heißt jedem $x \in {}^{\omega}\omega$ wird genau die reelle Zahl zugeordnet, die R^x kodiert. Dann ist f stetig, denn falls $x \restriction n = y \restriction n$, dann stimmen T_x und T_y für Elemente mit Länge bis zu n überein, und somit gilt $R^x \restriction n = R^y \restriction n$, das heißt $f(x) \restriction n = f(y) \restriction n$.
Es bleibt zu zeigen, dass $x \in A$ genau dann, wenn $f(x) \in$ WO. Sei also $x \notin A$, dann ist $(T_x, \supset)$ nicht fundiert, also gibt es eine Folge $\langle s_i : i \in \omega \rangle$ mit $s_{i+1} \supset s_i$ und $s_i \in T_x$ für alle $i \in \omega$. Dann muss es aber auch eine Folge $\langle n_i : i \in \omega \rangle$ geben mit $n_{i+1} R_{f(x)} n_i$ für alle $i \in \omega$, also entspricht $f(x)$ einer nicht fundierten Relation.
Sei umgekehrt $f(x) \notin$ WO. Dann gibt es eine Folge $\langle n_i : i \in \omega \rangle$ von natürlichen Zahlen, sodass $n_{i+1} R^x n_i$ für alle $i \in \omega$. Es gilt $e(n_i) \in T_x$ für alle $i \in \omega$, denn angenommen es gäbe ein $e(n_k) \notin T_x$, dann wären auch alle $e(n_i)$ mit $i > k$ nicht in T_x, und die n_i mit $i > k$ würden eine unendliche absteigende Kette bezüglich $<$ bilden - ein Widerspruch, da $(\omega, <)$ eine Wohlordnung ist.
Man kann sehen, dass die endlichen Folgen $e(n_i)$ beliebig lang werden müssen,

darüber hinaus gilt sogar die folgende Aussage:

$$\forall k\ \exists i(k)\ \forall i \geq i(k)\ (lh(e(n_i)) \geq k \wedge e(n_i) \restriction k = e(n_{i(k)}) \restriction k)$$

Wählt man die $i(k)$ außerdem so, dass $i(k+1) > i(k)$ für alle $k \in \omega$, dann ist $\langle e(n_{i(k)}) \restriction k : k \in \omega \rangle$ eine Folge, die absteigend bezüglich $\supset$ ist. Das bedeutet aber, dass $(T_x, \supset)$ nicht fundiert ist, und daher $x \notin A$. □

Bemerkung. Der Beweis zeigt auch, dass WF $\mathbf{\Pi}^1_1$-vollständig ist.

4.3 Weitere Beobachtungen zur Entscheidbarkeit

Nachdem bereits gezeigt wurde, dass WO in unendlicher Zeit entscheidbar ist, kann man aus Abschnitt 4.2 sofort das folgende Resultat ableiten:

Korollar 4.9. *Jede $\mathbf{\Pi}^1_1$-Menge und jede $\mathbf{\Sigma}^1_1$-Menge ist in unendlicher Zeit entscheidbar.*

Beweis. Sei A eine beliebige $\mathbf{\Pi}^1_1$-Menge, und f die Funktion aus dem vorigen Beweis, sodass $x \in A$ genau dann, wenn $f(x) \in$ WO. f ist in unendlicher Zeit aus A berechenbar: Wenn der zu A gehörige Baum T bekannt ist (z.B. auf einem zusätzlichen Band der Maschine kodiert), kann für jedes $x \in A$ auch die Relation R^x bestimmt werden. Schließlich findet man die reelle Zahl, die R^x kodiert, und erhält $f(x)$.
Man betrachte also eine Ordinalzahl-Turingmaschine, die für jeden Input x zunächst $f(x)$ berechnet, und anschließend nach dem im Beweis von Satz 4.6 beschriebenen Verfahren entscheidet, ob $f(x) \in$ WO gilt.
Da jede $\mathbf{\Sigma}^1_1$-Menge das Komplement einer $\mathbf{\Pi}^1_1$-Menge ist, kann man klarerweise die gleiche Methode anwenden. □

Satz 4.10. *Eine Menge ist genau dann arithmetisch, wenn sie von einer Ordinalzahl-Turingmaschine mit einer beschränkten endlichen Anzahl an Limesschritten entscheidbar ist.*

Beweis. Im Beweis von Satz 4.3 kann man nachvollziehen, dass jede Σ^0_n Menge durch eine Berechnung mit höchstens n Limiten entscheidbar ist. Da jede arithmetische Formel in Σ^0_n für mindestens ein $n \in \omega$ ist, sind damit alle arithmetischen Mengen abgedeckt.
Sei umgekehrt eine Menge A vorgegeben, die von einem Programm p einer Ordinalzahl-Turingmaschine entschieden wird. Sei $\varphi_{p,\alpha}(x)$ der Inhalt des Ausgabebandes nach genau α Schritten der Berechnung p mit Input x, und $\varphi_{p,\alpha}(x)(i)$ der Inhalt des i-ten Feldes. Weiters sei $s(i)$ das i-te Zeichen eines endlichen Binärstrings s.

Behauptung: Die Relation „s ist ein Anfangsstück von $\varphi_{p,\omega\cdot n}(x)$“ (Notation: $s \subset \varphi_{p,\omega\cdot n}(x)$) ist arithmetisch für alle $n \in \omega$.
Die Behauptung kann mit Induktion gezeigt werden. Für $n = 0$ stimmt sie offensichtlich, da das Outputband zu Beginn der Berechnung nur Nullen enthält. Für den Induktionsschritt, sei p' ein abgewandeltes Programm, das gleich zu Beginn der Berechnung das gleiche tut, wie p im Limeszustand. (Man könnte p' zum Beispiel so definieren, dass die Maschine im nullten Schritt in den Limeszustand wechselt, und ab dann dieselben Anweisungen wie p befolgt.) Nun kann ein Feld im Schritt $\omega \cdot (n + 1)$ dann und nur dann 0 sein, wenn die Konfiguration zum Zeitpunkt $\omega \cdot n$ dazu führt, dass das Feld ab einem bestimmten Zeitpunkt konstant 0 ist. Also ist für einen endlichen String s der Länge $l(s)$ die Relation $s \subset \varphi_{p,\omega\cdot(n+1)}(x)$ äquivalent dazu, dass für alle $i < l(s)$ gilt:

$$s(i) = 0 \leftrightarrow \exists N \forall m > N \forall t \, [t \subset \varphi_{p,\omega\cdot n}(x) \wedge l(t) > m \rightarrow \varphi_{p',m}(t)(i) = 0]$$

Der Ausdruck $\forall t$ [...] ist äquivalent zu $\varphi_{p,\omega\cdot n+m}(x)(i) = 0$, da nur t betrachtet werden, die auf den ersten m Zeichen mit x übereinstimmen, und eine Ordinalzahl-Turingmaschine innerhalb von m Schritten nach einem Limes höchstens die ersten m Zeichen einlesen kann. Daher ist die Formel nach Induktionsvoraussetzung arithmetisch, und die Behauptung ist bewiesen.
Damit ist alles gezeigt, denn falls A mit höchstens n vielen Limiten entscheidbar ist, dann gilt $x \in A$ genau dann, wenn im Schritt $\omega \cdot n$ der Berechnung p auf dem Outputband 1 steht, das heißt $1 \subset \varphi_{p,\omega\cdot n}(x)$. Also ist A arithmetisch. □

Literatur

[1] Sanjeev Arora und Boaz Barak. *Computational Complexity. A Modern Approach*. Cambridge University Press, 2009.

[2] Joel David Hamkins und Andy Lewis. „Infinite Time Turing Machines“. In: *The Journal of Symbolic Logic* 65.2 (Juni 2000), S. 567–604.

[3] Ulrich Hedtstück. *Einführung in die Theoretische Informatik. Formale Sprachen und Automatentheorie*. 5. Auflage. Oldenbourg Wissenschaftsverlag, 2012.

[4] Thomas Jech. *Set Theory*. 3. Auflage. Springer Verlag, 2002.

[5] Akihiro Kanamori. *The Higher Infinite. Large Cardinals in Set Theory from Their Beginnings*. 2.Auflage. Springer Verlag, 2009.

[6] Alexander S. Kechris. *Classical Descriptive Set Theory*. Springer Verlag, 1995.

[7] Kenneth Kunen. *Set Theory. An Introduction to Independence Proofs*. North-Holland Publishing Company, 1980.

[8] Ralf Schindler. *Set Theory. Exploring Independence and Truth*. Springer Verlag, 2014.