

Superturingmaschinen

Ingo Blechschmidt

Curry Club Augsburg

4. Oktober 2016

1 Erinnerungen

- Gewöhnliche Turingmaschinen
- Ordinalzahlen

2 Grundlagen zu Superturingmaschinen

- Erste Schritte
- Fähigkeiten von Superturingmaschinen
- Laufzeit von Superturingmaschinen

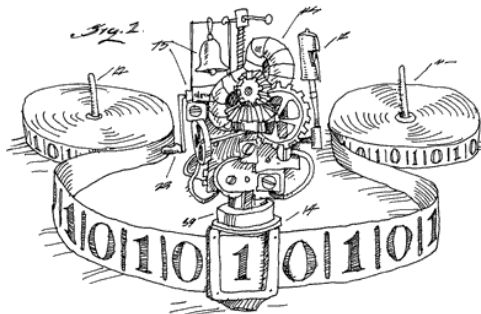
3 Besondere Phänomene

- Ausbrechen aus Wiederholungen
- Stempelbare Ordinalzahlen
- Lost-Melody-Theorem

4 Der effektive Topos

- Mathematische Alternativuniversen
- Das Wunder intuitionistischer Logik
- Effektive Bedeutung klassischer Tautologien

Ein Hoch auf Turingmaschinen



- | | |
|----------------------------------|---|
| 1 Schlichtheit | 4 Äquivalenz zu anderen Modellen |
| 2 Mechanischer Bezug | 5 Querverbindungen |
| 3 Robustheit des Konzepts | |

Lustiges zu Turingmaschinen

- 1 Schon kleine Turingmaschinen sind diffizil.
- 2 Es gibt Turingmaschinen, deren Halteverhalten unabhängig von Standard-Axiomen der Mathematik ist.

Lustiges zu Turingmaschinen

- 1 Schon kleine Turingmaschinen sind diffizil.
- 2 Es gibt Turingmaschinen, deren Halteverhalten unabhängig von Standard-Axiomen der Mathematik ist.
- 3 Alle sinnvollen Modelle für Berechenbarkeit stimmen für Funktionen $\mathbb{N} \rightarrow \mathbb{N}$ überein.

Lustiges zu Turingmaschinen

- 1 Schon kleine Turingmaschinen sind diffizil.
- 2 Es gibt Turingmaschinen, deren Halteverhalten unabhängig von Standard-Axiomen der Mathematik ist.
- 3 Alle sinnvollen Modelle für Berechenbarkeit stimmen für Funktionen $\mathbb{N} \rightarrow \mathbb{N}$ überein.
- 4 Eine Menge ist genau dann rekursiv aufzählbar, wenn sie durch eine Σ_1 -Aussage definierbar ist:

$$\{n \in \mathbb{N} \mid \text{es gibt } m \in \mathbb{N} \text{ mit } \heartsuit\},$$

Lustiges zu Turingmaschinen

- 1 Schon kleine Turingmaschinen sind diffizil.
- 2 Es gibt Turingmaschinen, deren Halteverhalten unabhängig von Standard-Axiomen der Mathematik ist.
- 3 Alle sinnvollen Modelle für Berechenbarkeit stimmen für Funktionen $\mathbb{N} \rightarrow \mathbb{N}$ überein.
- 4 Eine Menge ist genau dann rekursiv aufzählbar, wenn sie durch eine Σ_1 -Aussage definierbar ist:

$$\{n \in \mathbb{N} \mid \text{es gibt } m \in \mathbb{N} \text{ mit } \heartsuit\},$$

und wenn sie diophantisch ist:

$$\{n \in \mathbb{N} \mid \text{die Gl. } f(n, x_1, \dots, x_m) = 0 \text{ besitzt eine Lösung}\},$$

wobei f ein Polynom mit ganzzahligen Koeffizienten ist.

Ordinalzahlen messen Anordnung

3:    

W+L: 

#

 $1 + \omega:$ $W + Z:$

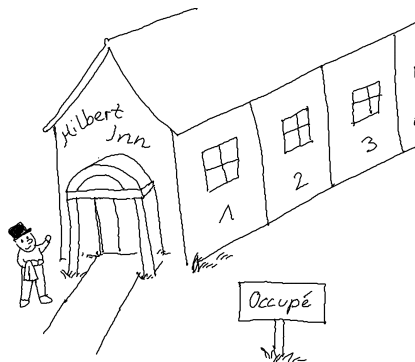
$\omega + \omega = 2\omega$:   

$2w+1$: 

$$3\omega, 4\omega, \dots, \omega^2, \dots, \omega^3, \dots$$

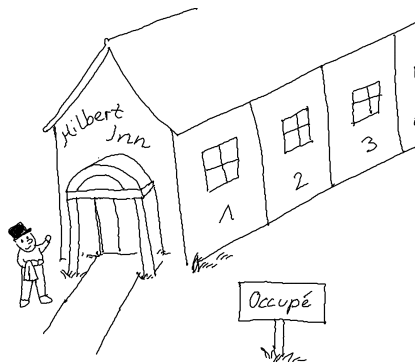
$$\omega, \dots, \omega^2, \dots, \omega^3, \dots$$

Kardinalzahlen messen Anzahl



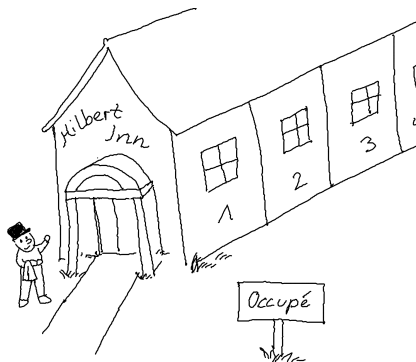
- Es gibt \aleph_0 viele natürliche Zahlen.

Kardinalzahlen messen Anzahl



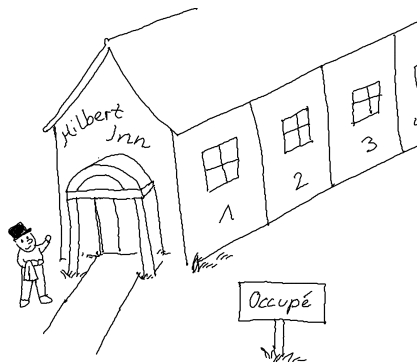
- Es gibt \aleph_0 viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0$,

Kardinalzahlen messen Anzahl



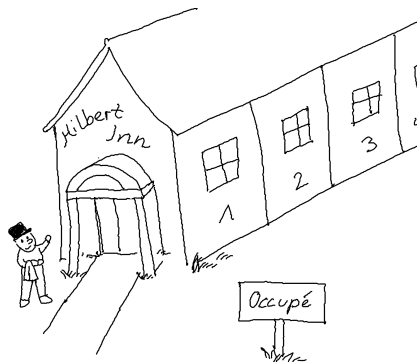
- Es gibt \aleph_0 viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0$, $\aleph_0 + \aleph_0 = \aleph_0$,

Kardinalzahlen messen Anzahl



- Es gibt \aleph_0 viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0$, $\aleph_0 + \aleph_0 = \aleph_0$, $\aleph_0 \cdot \aleph_0 = \aleph_0$.

Kardinalzahlen messen Anzahl



- Es gibt \aleph_0 viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0$, $\aleph_0 + \aleph_0 = \aleph_0$, $\aleph_0 \cdot \aleph_0 = \aleph_0$.
- Es gibt mehr als \aleph_0 viele reelle Zahlen.

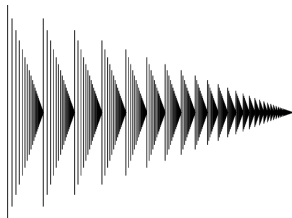
Was sind Superturingmaschinen?

Bei Superturingmaschinen ist die Zeitachse spannender:

- normal: $0, 1, 2, \dots$
- super: $0, 1, 2, \dots, \omega, \omega + 1, \dots, 2\omega, 2\omega + 1, \dots$

Wird eine Limesordinalzahl erreicht, so wird

- die Maschine in einen designierten Zustand versetzt,
- der Schreib-/Lesekopf auf den Anfang bewegt und
- der „lim sup“ aller vorherigen Bandinhalte genommen.



Was können Superturingmaschinen?

- Alles, was gewöhnliche Turingmaschinen können.
- Zahlentheoretische Behauptungen überprüfen:
 - \forall – „Für alle Zahlen gilt ...“
 - \exists – „Es gibt eine Zahl mit ...“
 - $\forall \exists$ – „Für alle Zahlen n gibt es jeweils eine Zahl m mit ...“
 - $\exists \forall$ – „Es gibt eine Zahl n , sodass für alle Zahlen m gilt: ...“
 - $\forall \exists \forall, \exists \forall \exists, \dots$
- Entscheiden, ob gewöhnliche Turingmaschinen halten.
- Superturingmaschinen und verwandte Maschinen emulieren.
- Π_1^1 - und Σ_1^1 -Aussagen entscheiden.

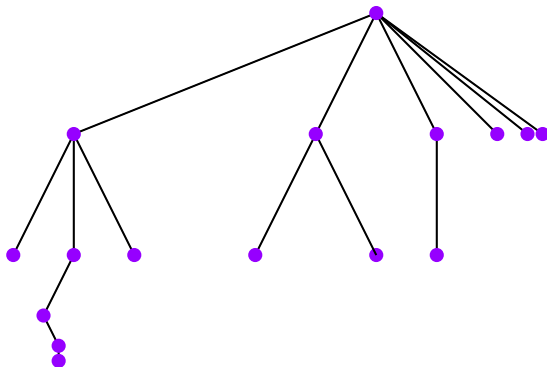
Was können Superturingmaschinen?

- Alles, was gewöhnliche Turingmaschinen können.
- Zahlentheoretische Behauptungen überprüfen:
 - \forall – „Für alle Zahlen gilt ...“
 - \exists – „Es gibt eine Zahl mit ...“
 - $\forall \exists$ – „Für alle Zahlen n gibt es jeweils eine Zahl m mit ...“
 - $\exists \forall$ – „Es gibt eine Zahl n , sodass für alle Zahlen m gilt: ...“
 - $\forall \exists \forall, \exists \forall \exists, \dots$
- Entscheiden, ob gewöhnliche Turingmaschinen halten.
- Superturingmaschinen und verwandte Maschinen emulieren.
- Π_1^1 - und Σ_1^1 -Aussagen entscheiden.

Aber: Superturingmaschinen können nicht alle Funktionen berechnen und nicht jede 0/1-Folge aufs Band schreiben.

Wohlfundierung von Bäumen

Ein Baum ist genau dann **wohlfundiert**, wenn er keinen unendlichen Pfad enthält.



Superturingmaschinen können die Wohlfundiertheit von Bäumen entscheiden.

Ein kleines Wunder

Superturingmaschinen können Π_1^1 - und Σ_1^1 -Aussagen entscheiden:

„Für jede Funktion $\mathbb{N} \rightarrow \mathbb{N}$ gilt ...“

„Es gibt eine Funktion $\mathbb{N} \rightarrow \mathbb{N}$ mit ...“

Wann halten Superturingmaschinen?

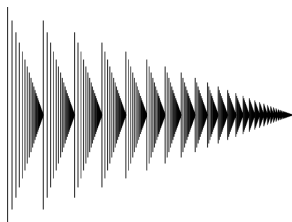
Schon nach **abzählbar vielen** (\aleph_0 vielen) Schritten hält jede Superturingmaschine entweder an oder wiederholt sich.

Wann halten Superturingmaschinen?

Schon nach **abzählbar vielen** (\aleph_0 vielen) Schritten hält jede Superturingmaschine entweder an oder wiederholt sich.

Sprechweise. Eine Ordinalzahl ist genau dann **abzählbar**, wenn sie nur abzählbar viele Vorgänger hat.

Genau die abzählbaren Ordinalzahlen lassen sich in \mathbb{R} einbetten.



Notation. Es ist ω_1 die erste Ordinalzahl, vor der überabzählbar unendlich viele Ordinalzahlen kommen.

Beweis. Angenommen, eine Superturingmaschine hat vor Schritt ω_1 noch nicht gehalten. Dann gibt es eine Ordinalzahl $\alpha_0 < \omega_1$, zu der sich alle Zellen, die sich bis vor ω_1 stabilisieren werden, schon stabilisiert haben. Ferner gibt es Ordinalzahlen

$$\alpha_0 < \alpha_1 < \alpha_2 < \cdots < \omega_1,$$

sodass sich zwischen α_n und α_{n+1} all die Zellen, die sich bis ω_1 noch ändern werden, jeweils mindestens einmal ändern. Sei $\delta = \lim_{n \rightarrow \infty} \alpha_n$. Dann ist die Aufnahme der Superturingmaschine bei δ gleich der bei ω_1 . Es ist $\delta < \omega_1$.

Was macht folgende Superturingmaschine?

Prüfe im Start- und Limeszustand, ob die aktuelle Zelle eine Eins enthält.

- Wenn ja, dann halte.
- Wenn nein, dann lass die Zelle aufleuchten und laufe ohne Unterlass nach rechts.

Ausbrechen aus Wiederholungen

Was macht folgende Superturingmaschine?

Prüfe im Start- und Limeszustand, ob die aktuelle Zelle eine Eins enthält.

- Wenn ja, dann halte.
- Wenn nein, dann lass die Zelle aufleuchten und laufe ohne Unterlass nach rechts.

Sie scheint sich zu wiederholen, hält aber nach Schritt ω^2 .

Ausbrechen aus Wiederholungen

Was macht folgende Superturingmaschine?

Prüfe im Start- und Limeszustand, ob die aktuelle Zelle eine Eins enthält.

- Wenn ja, dann halte.
- Wenn nein, dann lass die Zelle aufleuchten und laufe ohne Unterlass nach rechts.

Sie scheint sich zu wiederholen, hält aber nach Schritt ω^2 .

Eine Superturingmaschine wiederholt sich genau dann, wenn

- die Aufnahmen zu zwei Limesordinalzeiten gleich sind und
- zwischen diesen Zeiten keine Zellen, die Null waren, zu Eins werden.

Stempelbare Ordinalzahlen

Eine Ordinalzahl α ist genau dann **stempelbar** (clockable), falls es eine Superturingmaschine gibt, die genau nach Schritt α hält.

- Jede endliche Ordinalzahl ist stempelbar.
- Stempelbar sind außerdem: ω , 2ω , ω^2
- Sind α und β stempelbar, so auch $\alpha + \beta$ und $\alpha \cdot \beta$.
- Nur abzählbar viele Ordinalzahlen sind stempelbar.
- Jede rekursive Ordinalzahl ist stempelbar.

Stempelbare Ordinalzahlen

Eine Ordinalzahl α ist genau dann **stempelbar** (clockable), falls es eine Superturingmaschine gibt, die genau nach Schritt α hält.

Beschleunigungssatz

Ist $\alpha + n$ stempelbar, so auch α .

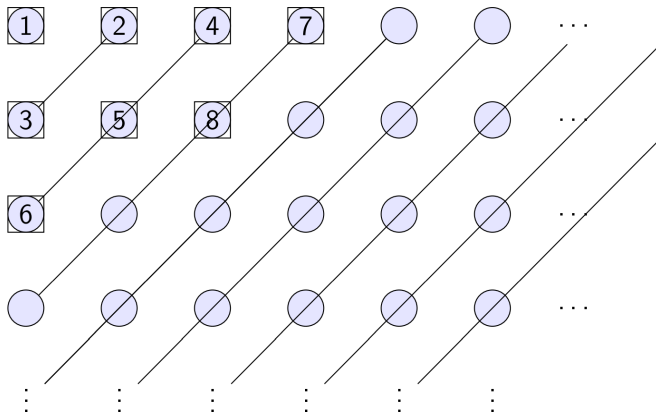
Große-Lücken-Satz

Für jede stempelbare Ordinalzahl α gibt eine Lücke der Länge $\geq \alpha$ in den stempelbaren Ordinalzahlen.

Viele-Lücken-Satz

Ist α eine schreibbare Ordinalzahl, so gibt es mindestens α viele Lücken der Länge $\geq \alpha$ in den stempelbaren Ordinalzahlen.

Erinnerung: Diagonalisierung



Lückenexistenzsatz

Die erste Lücke nach jeder stempelbaren Ordinalzahl hat Länge ω .

Beweis. Sei α eine stempelbare Ordinalzahl. Sei β die kleinste nicht-stempelbare Ordinalzahl nach α . Dann gibt es keine stempelbaren Ordinalzahlen zwischen β und $\beta + \omega$. Und $\beta + \omega$ selbst ist stempelbar durch folgendes Programm:

Emuliere alle Superturingmaschinen auf verzahnte Art und Weise. Behalte dabei insbesondere das Programm im Auge, das nach Schritt α halten wird. Sobald dieses gehalten hat, emuliere so lange weiter, bis der Zeitpunkt β erreicht ist, zu dem keine Superturingmaschine hält, und halte dann.

Zur Erkennung waren aber noch ω Schritte nötig.

Lost-Melody-Theorem

Es gibt Bandinhalte, die

- Superturingmaschinen nicht schreiben, aber
- erkennen können.

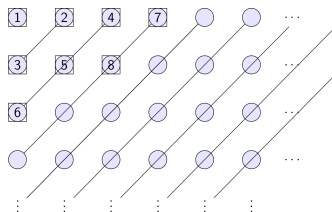
Lost-Melody-Theorem

Es gibt Bandinhalte, die

- Superturingmaschinen nicht schreiben, aber
- erkennen können.

Beweis. Sei c eine Kodierung aller Ablauffolgen aller Superturingmaschinen als unendliche 0/1-Folge.

- Dann ist c nicht schreibbar.
- Aber c ist erkennbar.



Mathematische Alternativuniversen

Zu jedem Rechenmodell \mathcal{M} gibt es einen **Topos** $\text{Eff}(\mathcal{M})$, in den wir mit **Realisierbarkeitstheorie** hineinschauen können.

Mathematische Alternativuniversen

Zu jedem Rechenmodell \mathcal{M} gibt es einen **Topos** $\text{Eff}(\mathcal{M})$, in den wir mit **Realisierbarkeitstheorie** hineinschauen können.

$\text{Eff}(\text{TM}) \models$ „Für jede Zahl n gibt es eine Primzahl $p > n$.“
bedeutet:

Es gibt eine Turingmaschine, die eine Zahl n vom Band einliest und eine Primzahl $p > n$ als Ausgabe aufs Band schreibt.

$\text{Eff}(\text{TM}) \models$ „Jede Zahl besitzt eine Primfaktorzerlegung.“
bedeutet:

Es gibt eine Turingmaschine, die eine Zahl n vom Band einliest und eine Liste von Primzahlen, deren Produkt n ist, aufs Band schreibt.

Was gilt in Alternativuniversen?

Metatheorem: Jede Aussage, die sich **intuitionistisch** beweisen lässt, gilt in allen Topoi.

Schon gewusst?

Intuitionistische Logik ist wie klassische Logik, nur ohne:

- Axiom vom ausgeschlossenen Dritten (LEM): $\varphi \vee \neg\varphi$
- Axiom der Doppelnegationselimination (DNE): $\neg\neg\varphi \Rightarrow \varphi$

So sind Widerspruchsbeweise nicht pauschal möglich.

Was gilt in Alternativuniversen?

Metatheorem: Jede Aussage, die sich **intuitionistisch** beweisen lässt, gilt in allen Topoi.

Schon gewusst?

Intuitionistische Logik ist wie klassische Logik, nur ohne:

- Axiom vom ausgeschlossenen Dritten (LEM): $\varphi \vee \neg\varphi$
- Axiom der Doppelnegationselimination (DNE): $\neg\neg\varphi \Rightarrow \varphi$

So sind Widerspruchsbeweise nicht pauschal möglich.

$\text{Eff}(\text{TM}) \models \varphi \vee \neg\varphi$

bedeutet:

Es gibt eine Turingmaschine, die entweder einen Zeugen von φ oder einen Zeugen von $\neg\varphi$ berechnet.

LEM für Gleichheit von Funktionen

$\text{Eff}(\text{TM}) \models$ „Für jede Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ gilt: Entweder ist f die Nullfunktion oder nicht.“

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M , welche eine Funktion $\mathbb{N} \rightarrow \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann entscheidet, ob M stets Null als Ausgabe produziert oder nicht.

Das stimmt nicht.

LEM für Gleichheit von Funktionen

$\text{Eff}(\text{TM}) \models$ „Für jede Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ gilt: Entweder ist f die Nullfunktion oder nicht.“

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M , welche eine Funktion $\mathbb{N} \rightarrow \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann entscheidet, ob M stets Null als Ausgabe produziert oder nicht.

Das stimmt nicht.

In $\text{Eff}(\text{STM})$ stimmt die Aussage.

LEM fürs Halten von Turingmaschinen

$\text{Eff}(\text{TM}) \models$ „Jede Turingmaschine M hält oder hält nicht.“

bedeutet:

Es gibt eine Turingmaschine, die die Kodierung einer Turingmaschine M als Eingabe vom Band liest und dann entscheidet, ob M hält oder nicht.

Das stimmt nicht.

Markovs Prinzip

$\text{Eff}(\text{TM}) \models$ „Für jede Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$, welche nicht die Nullfunktion ist, gibt es eine Stelle $n \in \mathbb{N}$ mit $f(n) \neq 0$.“

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M , welche eine Funktion $\mathbb{N} \rightarrow \mathbb{N}$ und zwar nicht die Nullfunktion berechnet, als Eingabe vom Band liest und dann eine Zahl n aufs Band schreibt, sodass M bei Eingabe von n nicht Null aufs Band schreibt.

Das stimmt!

Church–Turing-These

Die **Church–Turing-These** besagt: Lässt sich eine Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ in der „realen Welt berechnen“, so gibt es eine Turingmaschine, die f berechnet.

$\text{Eff}(\text{TM}) \models$ „Jede Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ lässt sich durch eine Turingmaschine berechnen.“

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M , welche eine Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann die Kodierung einer Turingmaschine, welche f berechnet, aufs Band schreibt.

Das ist trivial!

Church–Turing-These

Die **Church–Turing-These** besagt: Lässt sich eine Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ in der „realen Welt berechnen“, so gibt es eine Turingmaschine, die f berechnet.

$\text{Eff}(\text{TM}) \models$ „Jede Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ lässt sich durch eine Turingmaschine berechnen.“

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M , welche eine Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ berechnet, als Eingabe vom Band liest und dann die Kodierung einer Turingmaschine, welche f berechnet, aufs Band schreibt.

Das ist trivial!

In $\text{Eff}(\lambda)$ stimmt die Aussage nicht.