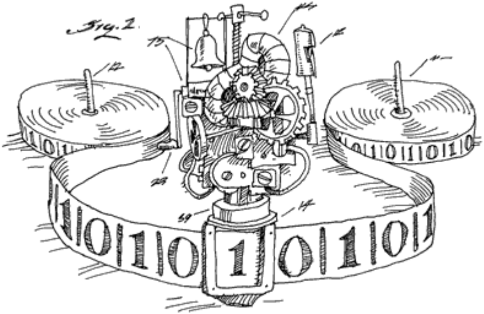
Superturingmaschinen

Ingo Blechschmidt

Curry Club Augsburg

4. Oktober 2016 und 3. November 2016

Ein Hoch auf Turingmaschinen



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

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 $\mathbb{N} \to \mathbb{N}$ 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 $\mathbb{N} \to \mathbb{N}$ 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 $\mathbb{N} \to \mathbb{N}$ überein.
4. Eine Menge ist genau dann rekursiv aufzählbar, wenn sie durch eine Σ_1-Aussage definierbar ist:

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

und wenn sie diophantisch ist:

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

wobei f ein Polynom mit ganzzahligen Koeffizienten ist.

Eine Σ_1-Aussage ist eine Aussage der Form

$$„\text{Es gibt } m \in \mathbb{N} \text{ mit } \heartsuit.“,$$

wobei in der Teilaussage $\heartsuit$ nur noch *beschränkte Quantifikation* vorkommen darf – also Formeln wie

$$„\text{Für alle Zahlen kleiner als } \cdots \text{ gilt } \ldots“$$

oder

$$„\text{Es gibt eine Zahl kleiner als } \cdots \text{ mit } \ldots“$$

und nicht Formeln wie

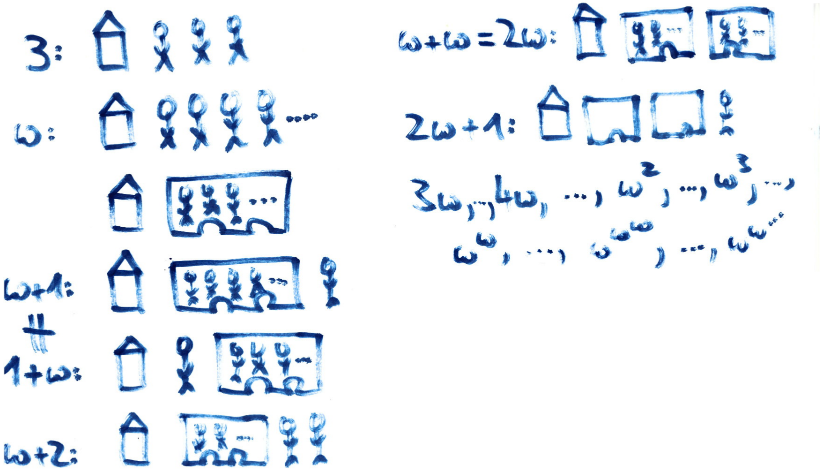
$$„\text{Für alle Zahlen gilt } \ldots“$$

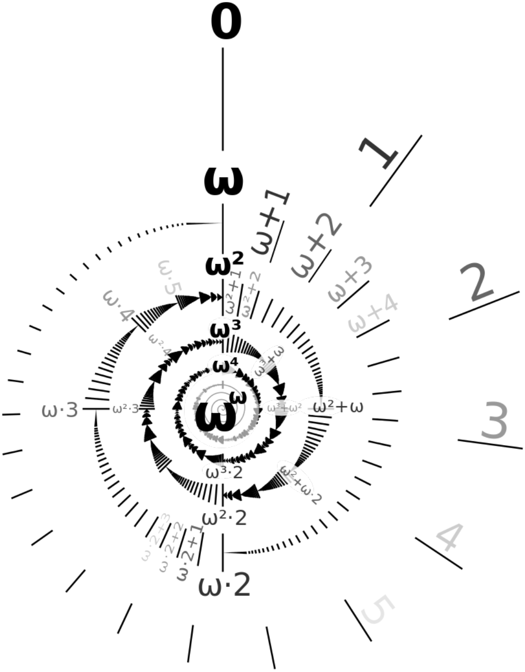
und

$$„\text{Es gibt eine Zahl mit } \ldots“.$$

Die Teilaussage $\heartsuit$ muss also in endlicher Zeit überprüfbar sein.

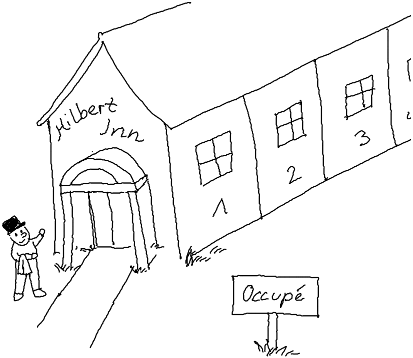
Ordinalzahlen messen Anordnung





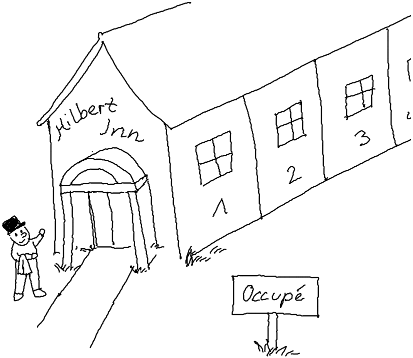
0
1
2
3
4
5
ω
ω+1
ω+2
ω+3
ω+4
ω·2
ω·2+1
ω·2+2
ω·2+3
ω·3
ω·4
ω·5
ω²
ω²+1
ω²+2
ω²+ω
ω²+ω·2
ω²·2
ω²·3
ω²·4
ω³
ω³+ω
ω³+ω²
ω³·2
ω⁴
ω^ω

Kardinalzahlen messen Anzahl



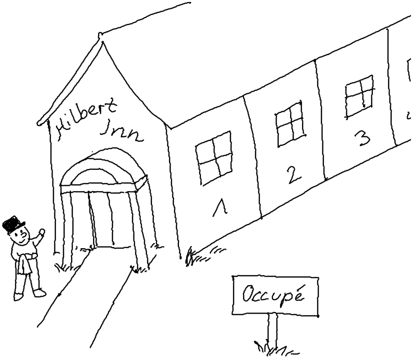
- Es gibt $\aleph_0$ viele natürliche Zahlen.

Kardinalzahlen messen Anzahl



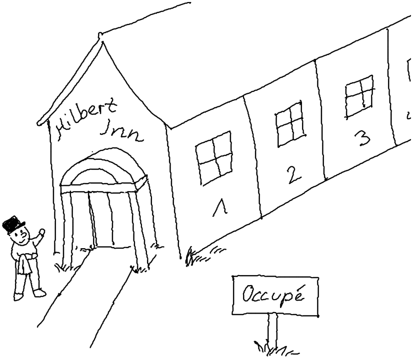
- Es gibt $\aleph_0$ viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0$,

Kardinalzahlen messen Anzahl



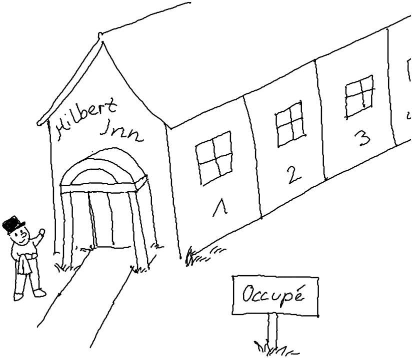
- Es gibt $\aleph_0$ viele natürliche Zahlen.
- $\aleph_0 + 1 = \aleph_0, \quad \aleph_0 + \aleph_0 = \aleph_0,$

Kardinalzahlen messen Anzahl



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

Kardinalzahlen messen Anzahl



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

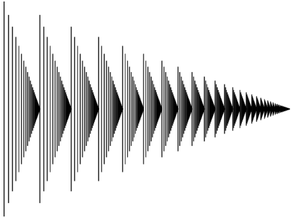
Was sind Superturingmaschinen?

Bei Superturingmaschinen ist die Zeitachse spannender:

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

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: …"
  - $\forall\exists\forall$, $\exists\forall\exists$, …
- Entscheiden, ob gewöhnliche Turingmaschinen halten.
- Superturingmaschinen und verwandte Maschinen simulieren.
- Π^1_1- und Σ^1_1-Aussagen entscheiden.

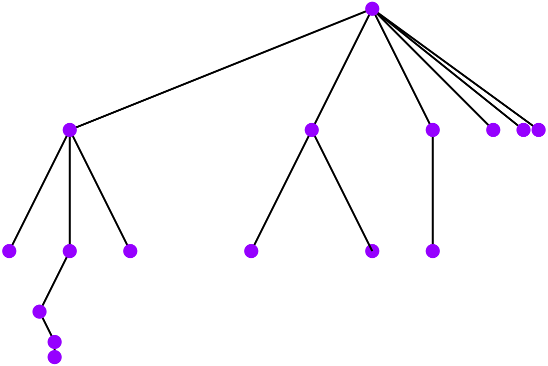
Was können Superturingmaschinen?

- 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: …“
  - $\forall\exists\forall$, $\exists\forall\exists$, …
- Entscheiden, ob gewöhnliche Turingmaschinen halten.
- Superturingmaschinen und verwandte Maschinen simulieren.
- 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.

Fundierung von Bäumen

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



Superturingmaschinen können die Fundiertheit von Bäumen entscheiden.

Ein kleines Wunder

Superturingmaschinen können Π^1_1- und Σ^1_1-Aussagen entscheiden:

„Für jede Funktion $\mathbb{N} \to \mathbb{N}$ gilt …“

„Es gibt eine Funktion $\mathbb{N} \to \mathbb{N}$ mit …“

Und das, obwohl es überabzählbar viele Funktionen $\mathbb{N} \to \mathbb{N}$ gibt, aber Superturingmaschinen nur ein abzählbares Band verwenden und (nächste Folie) immer schon nach abzählbar vielen Schritten halten oder in Endlosschleifen geraten.

Wann halten Superturingmaschinen?

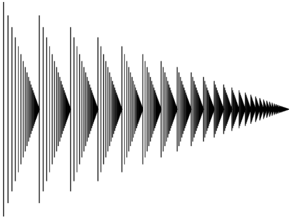
Schon nach **abzählbar vielen** ($\leq \aleph_0$ vielen) Schritten hält jede Superturingmaschine entweder an oder wiederholt sich.

Wann halten Superturingmaschinen?

Schon nach **abzählbar vielen** ($\leq \aleph_0$ 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.

Behauptung. Hat eine Superturingmaschine nach *abzählbar vielen* Schritten noch nicht angehalten, so hält sie nie.

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 $\delta = \lim_{n\to\infty} \alpha_n$. Das ist eine Ordinalzahl $< \omega_1$, also eine abzählbare Ordinalzahl. Dann ist die Aufnahme der Superturingmaschine bei δ gleich der bei ω_1. Die Superturingmaschine wiederholt sich also.

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.

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

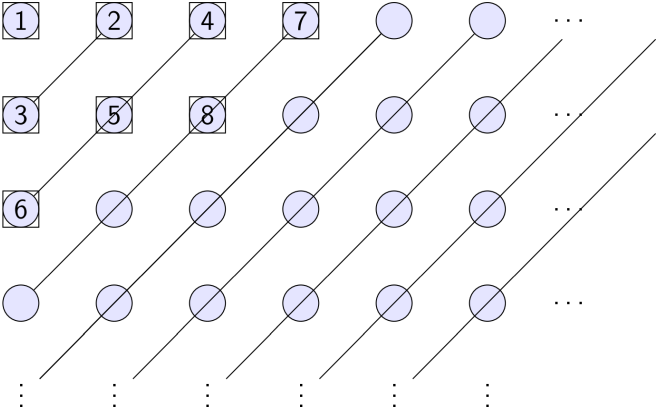
Für jede stempelbare Ordinalzahl α gibt es eine Lücke der Länge $\geq \alpha$ in den stempelbaren Ordinalzahlen.

Viele-Lücken-Satz

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

Kurioserweise gibt es auch den *Lückenlose-Blöcke-Satz* (Gapless Blocks Theorem): Es gibt in den Ordinalzahlen „lange Abschnitte" von lauter 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:

> Simuliere alle Superturingmaschinen auf verzahnte Art und Weise. Behalte dabei insbesondere das Programm im Auge, das nach Schritt α halten wird. Sobald dieses gehalten hat, simuliere 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.

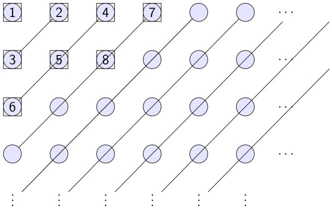
Lost-Melody-Theorem

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** $\mathrm{Eff}(\mathcal{M})$, in den wir mit **Realisierbarkeitstheorie** hineinschauen können.

Mathematische Alternativuniversen

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

$\mathrm{Eff}(\mathrm{TM}) \models$ „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.

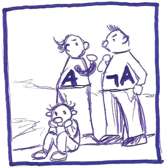
$\mathrm{Eff}(\mathrm{TM}) \models$ „Jede Zahl besitzt eine Primfaktorzerlegung."

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.

LEM für Gleichheit von Funktionen

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

bedeutet:

Es gibt eine Turingmaschine, die eine Kodierung einer Turingmaschine M, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ 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

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

bedeutet:

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

Das stimmt nicht.

Bemerkung: Aus externer Sicht ist das Objekt $\mathbb{N}^{\mathbb{N}}$ des effektiven Topos die Menge der **berechenbaren** Funktionen.

LEM für Gleichheit von Funktionen

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

bedeutet:

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

Das stimmt nicht.

Bemerkung: Aus externer Sicht ist das Objekt $\mathbb{N}^{\mathbb{N}}$ des effektiven Topos die Menge der **berechenbaren** Funktionen.

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

LEM fürs Halten von Turingmaschinen

$\mathrm{Eff}(\mathrm{TM}) \models$ „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.

LEM fürs Halten von Turingmaschinen

Eff(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.

In Eff(STM) stimmt die Aussage.

Markovs Prinzip

$\mathrm{Eff}(\mathrm{TM}) \models$ „Für jede Funktion $f : \mathbb{N} \to \mathbb{N}$, 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 $\mathbb{N} \to \mathbb{N}$ 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! (Unbeschränkte Suche.)

Church–Turing-These

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

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

bedeutet:

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

Das ist trivial! „cat“ ist die gesuchte Maschine.

Church–Turing-These

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

$\mathrm{Eff}(\mathrm{TM}) \models$ „Jede Funktion $f : \mathbb{N} \to \mathbb{N}$ lässt sich durch eine Turingmaschine berechnen."

bedeutet:

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

Das ist trivial! „cat" ist die gesuchte Maschine.

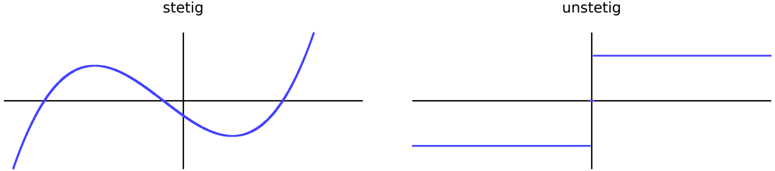
In $\mathrm{Eff}(\mathrm{STM})$ und $\mathrm{Eff}(\lambda)$ stimmt die Aussage nicht.

Automatische Stetigkeit

Im üblichen Universum stimmt folgende Aussage **nicht**:

„Jede Funktion $f : \mathbb{R} \to \mathbb{R}$ ist stetig."

Eine Funktion f heißt genau dann **stetig**, falls für jede Zahl x zur Bestimmung von endlich vielen Nachkommastellen von $f(x)$ schon endlich viele Nachkommastellen von x genügen.

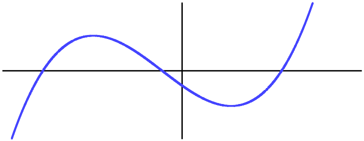


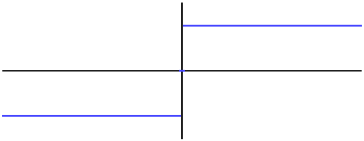
Automatische Stetigkeit

Im üblichen Universum stimmt folgende Aussage **nicht**:

„Jede Funktion $f : \mathbb{R} \to \mathbb{R}$ ist stetig.“

Eine Funktion f heißt genau dann **stetig**, falls für jede Zahl x zur Bestimmung von endlich vielen Nachkommastellen von $f(x)$ schon endlich viele Nachkommastellen von x genügen.





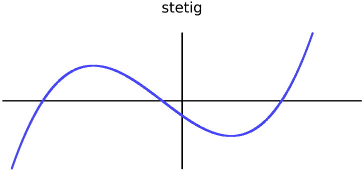
Stimmt in Eff(TM).

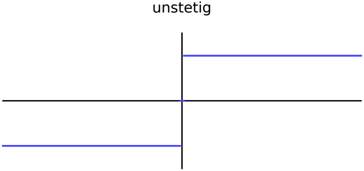
Automatische Stetigkeit

Im üblichen Universum stimmt folgende Aussage **nicht**:

„Jede Funktion $f : \mathbb{R} \to \mathbb{R}$ ist stetig."

Eine Funktion f heißt genau dann **stetig**, falls für jede Zahl x zur Bestimmung von endlich vielen Nachkommastellen von $f(x)$ schon endlich viele Nachkommastellen von x genügen.



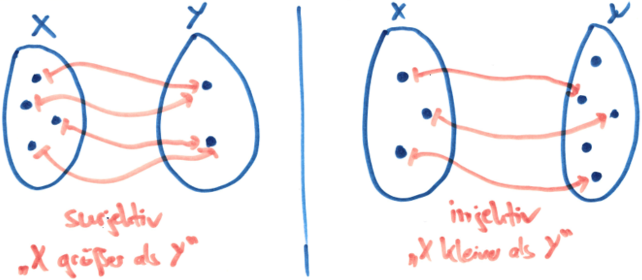


Stimmt in Eff(TM). Stimmt in Eff(RW), falls black boxes und private Kommunikationskanäle möglich sind und in endlicher Zeit nur endlich viele Rechenschritte ausgeführt werden können.

Seltsame Größenverhältnisse

Es gibt keine Surjektion $\mathbb{N} \to \mathbb{N}^{\mathbb{N}}$; die Menge $\mathbb{N}^{\mathbb{N}}$ der Funktionen $\mathbb{N} \to \mathbb{N}$ ist viel größer als $\mathbb{N}$.

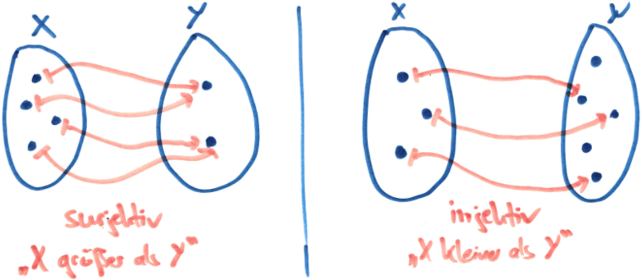
In klassischer Logik folgt: Es gibt auch keine Injektion $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$. Das drückt dieselbe Intuition über das Größenverhältnis aus.



Seltsame Größenverhältnisse

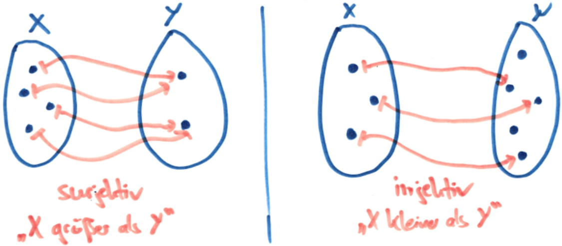
Es gibt keine Surjektion $\mathbb{N} \to \mathbb{N}^{\mathbb{N}}$; die Menge $\mathbb{N}^{\mathbb{N}}$ der Funktionen $\mathbb{N} \to \mathbb{N}$ ist viel größer als $\mathbb{N}$.

In klassischer Logik folgt: Es gibt auch keine Injektion $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$. Das drückt dieselbe Intuition über das Größenverhältnis aus.



Aber in Eff(STM) gibt es eine solche Injektion!

Seltsame Größenverhältnisse



Eff(STM) $\models$ „Es gibt eine Injektion $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$.“

bedeutet:

Es gibt eine Superturingmaschine, welche bei Eingabe einer Kodierung einer Superturingmaschine A, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ berechnet, eine Zahl $n(A)$ berechnet und aufs Band schreibt. Dabei darf nur dann $n(A) = n(B)$ sein, wenn A und B dieselbe Funktion berechnen.

Seltsame Größenverhältnisse

Die Superturingmaschine

> Lese die Kodierung einer Superturingmaschine A vom Band ein. Gehe nun alle natürlichen Zahlen n der Reihe nach durch und prüfe jeweils, ob die n-te Superturingmaschine dasselbe Verhalten zeigt wie A. Da A terminiert, ist das entscheidbar. Gebe die kleinste so gefundene Zahl n aus.

schreibt bei Eingabe einer Kodierung einer Superturingmaschine A, welche eine Funktion $\mathbb{N} \to \mathbb{N}$ berechnet, eine Zahl $n(A)$ aufs Band. Dabei ist nur dann $n(A) = n(B)$, wenn A und B dieselbe Funktion berechnen.