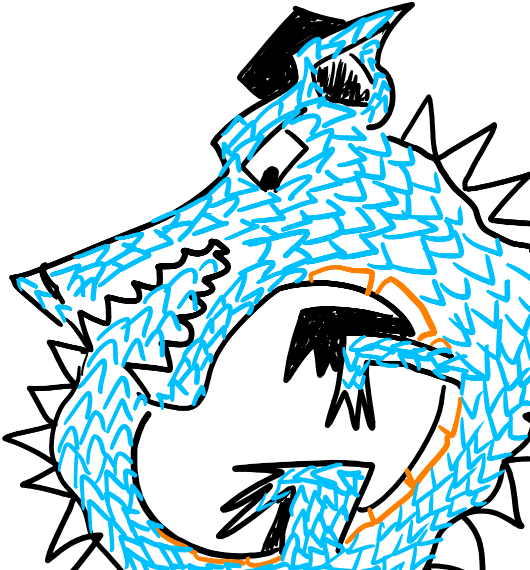
Unberechenbar!

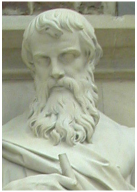
Clara Löh

Universität Regensburg

20|05|2025



Panorama

Prolog: Mathematik $\neq$ Rechnen

Probleme!

Lösungen?

Epilog: Proofs $=$ Programs

Prolog: Mathematatik $\neq$ Rechnen

Was ist Mathematik?

$$\left(\cos\sqrt{2025}\right)^2 + \left(\sin\sqrt{2025}\right)^2 - 1 = ?$$

Was ist Mathematik?

$$\left(\cos\sqrt{2025}\right)^2 + \left(\sin\sqrt{2025}\right)^2 - 1 = ?$$

Satz. Die Quadratur des Kreises ist mit Zirkel und Lineal nicht möglich.

Beweis. Angenommen, sie wäre möglich.

Insbesondere wäre dann $\sqrt{\pi}$ aus $\{0,1\}$ mit Zirkel und Lineal konstruierbar.

Also wäre $\sqrt{\pi}$ algebraisch. $\lightning$ $\square$

Was ist Mathematik? Nicht Rechnen!

Wissenschaft des abstrakten Denkens, insbesondere:

- abstrakte Strukturen (Zahlen, Graphen, Geometrie, ...)
- formale Methoden.

Was ist Mathematik? Nicht Rechnen!

Wissenschaft des abstrakten Denkens, insbesondere:

- abstrakte Strukturen (Zahlen, Graphen, Geometrie, ...)
- formale Methoden.

Bausteine:

- Axiome legen die Spielregeln fest
- Definitionen führen neue Begriffe ein
- Sätze formulieren Aussagen über mathematische Objekte
- Beweise sind formale Begründungen für behauptete Aussagen
- Beispiele liefern Anschauung und Anwendungen

Was ist Mathematik? Nicht Rechnen!

Wissenschaft des abstrakten Denkens, insbesondere:

- abstrakte Strukturen (Zahlen, Graphen, Geometrie, ...)
- formale Methoden.

Bausteine:

- Axiome legen die Spielregeln fest
- Definitionen führen neue Begriffe ein
- Sätze formulieren Aussagen über mathematische Objekte
- Beweise sind formale Begründungen für behauptete Aussagen
- Beispiele liefern Anschauung und Anwendungen

Jeder diese Bausteine ist präzise formalisiert.

Probleme!

Mathematische Probleme [Hilbert, 1900]

„Es ist schwierig und oft unmöglich, den Wert eines Problems im Voraus richtig zu beurteilen; denn schließlich entscheidet der Gewinn, den die Wissenschaft dem Problem verdankt. Dennoch können wir fragen, ob es allgemeine Merkmale giebt, die ein gutes mathematisches Problem kennzeichnen.

... Mathematische Probleme ...

Ein alter französischer Mathematiker hat gesagt: Eine mathematische Theorie ist nicht eher als vollkommen anzusehen, als bis du sie so klar gemacht hast, daß du sie dem ersten Manne erklären könntest, den du auf der Straße triffst.

. . . Mathematische Probleme . . .

Ein alter französischer Mathematiker hat gesagt: Eine mathematische Theorie ist nicht eher als vollkommen anzusehen, als bis du sie so klar gemacht hast, daß du sie dem ersten Manne erklären könntest, den du auf der Straße triffst.
Diese Klarheit und leichte Faßlichkeit, wie sie hier so drastisch für eine mathematische Theorie verlangt wird, möchte ich viel mehr von einem mathematischen Problem fordern, wenn dasselbe vollkommen sein soll; denn das Klare und leicht Faßliche zieht uns an, das Verwickelte schreckt uns ab.

. . . Mathematische Probleme . . .

Ein alter französischer Mathematiker hat gesagt: Eine mathematische Theorie ist nicht eher als vollkommen anzusehen, als bis du sie so klar gemacht hast, daß du sie dem ersten Manne erklären könntest, den du auf der Straße triffst.
Diese Klarheit und leichte Faßlichkeit, wie sie hier so drastisch für eine mathematische Theorie verlangt wird, möchte ich viel mehr von einem mathematischen Problem fordern, wenn dasselbe vollkommen sein soll; denn das Klare und leicht Faßliche zieht uns an, das Verwickelte schreckt uns ab.

Ein mathematisches Problem sei ferner schwierig, damit es uns reizt, und dennoch nicht völlig unzugänglich, damit es unserer Anstrengung nicht spotte; es sei uns ein Wahrzeichen auf den verschlungenen Pfaden zu verborgenen Wahrheiten – uns hernach lohnend mit der Freude über die gelungene Lösung.

. . . Mathematische Probleme

[. . .]

Diese merkwürdige Thatsache neben anderen philosophischen Gründen ist es wohl, welche in uns eine Ueberzeugung entstehen läßt, die jeder Mathematiker gewiß teilt, die aber bis jetzt wenigstens niemand durch Beweise gestutzt hat – ich meine die Ueberzeugung, daß ein jedes bestimmte mathematische Problem einer strengen Erledigung notwendig fähig sein müsse, sei es, daß es gelingt, die Beantwortung der gestellten Frage zu geben, sei es, daß die Unmöglichkeit seiner Lösung und damit die Notwendigkeit des Mißlingens aller Versuche dargethan wird.

. . . Mathematische Probleme

[. . .]

Diese merkwürdige Thatsache neben anderen philosophischen Gründen ist es wohl, welche in uns eine Ueberzeugung entstehen läßt, die jeder Mathematiker gewiß teilt, die aber bis jetzt wenigstens niemand durch Beweise gestutzt hat – ich meine die Ueberzeugung, daß ein jedes bestimmte mathematische Problem einer strengen Erledigung notwendig fähig sein müsse, sei es, daß es gelingt, die Beantwortung der gestellten Frage zu geben, sei es, daß die Unmöglichkeit seiner Lösung und damit die Notwendigkeit des Mißlingens aller Versuche dargethan wird.

[. . .]

Diese Ueberzeugung von der Lösbarkeit eines jeden mathematischen Problems ist uns ein kräftiger Ansporn während der Arbeit; wir hören in uns den steten Zuruf: *Da ist das Problem, suche die Lösung. Du kannst sie durch reines Denken finden; denn in der Mathematik giebt es, kein Ignorabimus!*"

Das zweite Hilbertsche Problem

Die Widerspruchslosigkeit der arithmetischen Axiome

„Wenn es sich darum handelt, die Grundlagen einer Wissenschaft zu untersuchen, so hat man ein System von Axiomen aufzustellen, welche eine genaue und vollständige Beschreibung derjenigen Beziehungen enthalten, die zwischen den elementaren Begriffen jener Wissenschaft stattfinden.

Das zweite Hilbertsche Problem

Die Widerspruchslosigkeit der arithmetischen Axiome

„Wenn es sich darum handelt, die Grundlagen einer Wissenschaft zu untersuchen, so hat man ein System von Axiomen aufzustellen, welche eine genaue und vollständige Beschreibung derjenigen Beziehungen enthalten, die zwischen den elementaren Begriffen jener Wissenschaft stattfinden. Die aufgestellten Axiome sind zugleich die Definitionen jener elementaren Begriffe und jede Aussage innerhalb des Bereiches der Wissenschaft, deren Grundlagen wir prüfen, gilt uns nur dann als richtig, falls sie sich mittelst einer endlichen Anzahl logischer Schlüsse aus den aufgestellten Axiomen ableiten läßt.

. . . Das zweite Hilbertsche Problem

Bei näherer Betrachtung entsteht die Frage, *ob etwa gewisse Aussagen einzelner Axiome sich untereinander bedingen und ob nicht somit die Axiome noch gemeinsame Bestandteile enthalten, die man beseitigen muß, wenn man zu einem System von Axiomen gelangen will, die völlig von einander unabhängig sind.*

... Das zweite Hilbertsche Problem

Bei näherer Betrachtung entsteht die Frage, *ob etwa gewisse Aussagen einzelner Axiome sich untereinander bedingen und ob nicht somit die Axiome noch gemeinsame Bestandteile enthalten, die man beseitigen muß, wenn man zu einem System von Axiomen gelangen will, die völlig von einander unabhängig sind.*

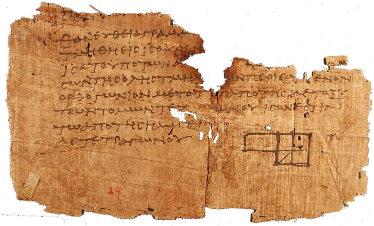
Vor Allem aber möchte ich unter den zahlreichen Fragen, welche hinsichtlich der Axiome gestellt werden können, dies als das wichtigste Problem bezeichnen, *zu beweisen, daß dieselben untereinander widerspruchslos sind, d.h. daß man auf Grund derselben mittelst einer endlichen Anzahl von logischen Schlüssen niemals zu Resultaten gelangen kann, die miteinander in Widerspruch stehen.*“

Drei Fragen über Axiomensysteme

- Minimalität

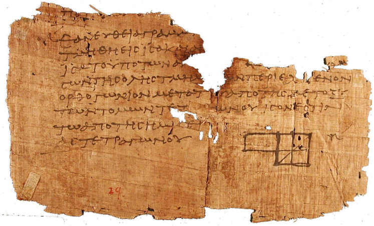
  Ist das Axiomensystem minimal?

  D.h. kann keines der Axiome aus den anderen abgeleitet werden?



Drei Fragen über Axiomensysteme

- Minimalität
  Ist das Axiomensystem minimal?
  D.h. kann keines der Axiome aus den anderen abgeleitet werden?



Cyl 1. The front and back inclusions assemble to an exact functor

$$\begin{array}{ccc} \mathrm{ArC} & \longrightarrow & F_1C \\ (A \xrightarrow[f]{} B) & \longmapsto & (A \vee B \xrightarrow[j_1 \vee j_2]{} T(f)) \ . \end{array}$$

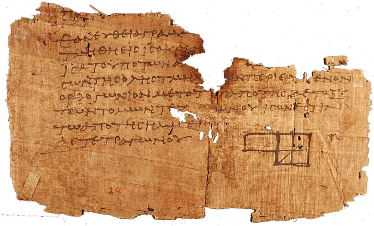
Cyl 2. $T(* \to A) = A$, for every $A \in C$, and the projection and back inclusion are the identity map on A.

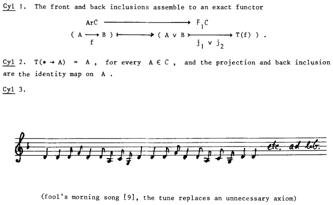
Cyl 3.

(fool's morning song [9], the tune replaces an unnecessary axiom)

Drei Fragen über Axiomensysteme

- **Minimalität**
  Ist das Axiomensystem minimal?
  D.h. kann keines der Axiome aus den anderen abgeleitet werden?





Cyl 1. The front and back inclusions assemble to an exact functor

$$\mathrm{Ar}\,C \longrightarrow F_1 C$$
$$(A \xrightarrow{f} B) \longmapsto (A \vee B \xrightarrow{j_1 \vee j_2} T(f)) .$$

Cyl 2. $T(* \to A) = A$, for every $A \in C$, and the projection and back inclusion are the identity map on A.

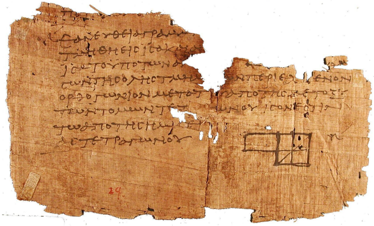
Cyl 3.

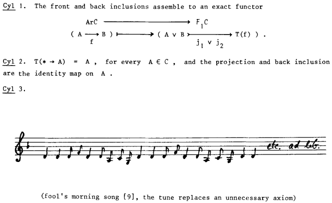
(fool's morning song [9], the tune replaces an unnecessary axiom)

- **Widerspruchsfreiheit** [Hilbert, 1900]
  Kann man beweisen, dass das Axiomensystem widerspruchsfrei ist?
  D.h. lässt sich keine Aussage der Form *A und nicht A* daraus ableiten?

Drei Fragen über Axiomensysteme

- **Minimalität**
  Ist das Axiomensystem minimal?
  D.h. kann keines der Axiome aus den anderen abgeleitet werden?





- **Widerspruchsfreiheit** [Hilbert, 1900]
  Kann man beweisen, dass das Axiomensystem widerspruchsfrei ist?
  D.h. lässt sich keine Aussage der Form *A und nicht A* daraus ableiten?
- **Entscheidungsproblem** [Hilbert, Ackermann; 1928]
  Gibt es ein effizientes Verfahren, das entscheiden kann,
  ob eine gegebene Aussage in der Situation des Axiomensystems
  aus den Axiomen abgeleitet werden kann oder nicht?

Lösungen?

Ignorabimus!

- Gödel [1931]
  Ist ein Axiomensystem ausdrucksstark genug,
  um die Arithmetik zu formalisieren,
  so kann es seine eigene Widerspruchsfreiheit *nicht* beweisen.

Ignorabimus!

- Gödel [1931]
  Ist ein Axiomensystem ausdrucksstark genug,
  um die Arithmetik zu formalisieren,
  so kann es seine eigene Widerspruchsfreiheit *nicht* beweisen.
- Church/Turing [1936]
  Das Entscheidungsproblem für die Logik erster Stufe ist *nicht* lösbar.
  Das Halteproblem für Turingmaschinen ist *nicht* lösbar.

Ignorabimus!

- Gödel [1931]
  Ist ein Axiomensystem ausdrucksstark genug,
  um die Arithmetik zu formalisieren,
  so kann es seine eigene Widerspruchsfreiheit *nicht* beweisen.
- Church/Turing [1936]
  Das Entscheidungsproblem für die Logik erster Stufe ist *nicht* lösbar.
  Das Halteproblem für Turingmaschinen ist *nicht* lösbar.

Frage

- Wie beweist man solche Aussagen?
- Warum sind diese Ergebnisse relevant?

Die Idee

Jedes System,

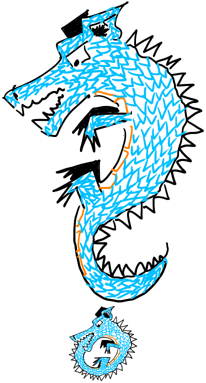
- das hinreichend komplexe Selbstbezüglichkeiten ausdrücken kann,
- kann keine allmächtigen Konstrukte besitzen,
- da sonst Widersprüche auftreten.

Die Idee

Jedes System,

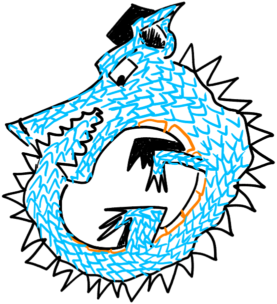
- das hinreichend komplexe Selbstbezüglichkeiten ausdrücken kann,
- kann keine allmächtigen Konstrukte besitzen,
- da sonst Widersprüche auftreten.

Häh?



Monster

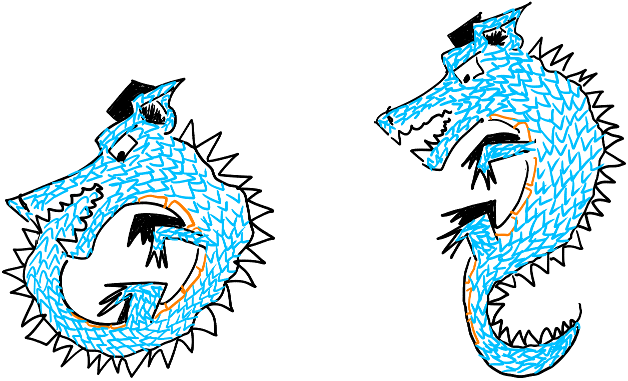
Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.



Monster

Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.

Frage. Gibt es ein Monster, das genau die Monster frisst,
die sich nicht selbst fressen?



Monster

Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.

Frage. Gibt es ein Monster, das genau die Monster frisst,
die sich nicht selbst fressen?

Antwort. Nein!

Monster

Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.

Frage. Gibt es ein Monster, das genau die Monster frisst,
die sich nicht selbst fressen?

Antwort. Nein!

Beweis.
Angenommen, es gäbe ein solches Monster M.

Monster

Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.

Frage. Gibt es ein Monster, das genau die Monster frisst, die sich nicht selbst fressen?

Antwort. Nein!

Beweis.
Angenommen, es gäbe ein solches Monster M.
Nach dem Axiom frisst sich M entweder selbst oder nicht.

Monster

Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.

Frage. Gibt es ein Monster, das genau die Monster frisst,
die sich nicht selbst fressen?

Antwort. Nein!

Beweis.
Angenommen, es gäbe ein solches Monster M.
Nach dem Axiom frisst sich M entweder selbst oder nicht.

① Falls sich M selbst frisst, so frisst sich M nach Definition nicht, denn M frisst genau die Monster, die sich selbst fressen.

Monster

Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.

Frage. Gibt es ein Monster, das genau die Monster frisst,
die sich nicht selbst fressen?

Antwort. Nein!

Beweis.
Angenommen, es gäbe ein solches Monster M.
Nach dem Axiom frisst sich M entweder selbst oder nicht.

① Falls sich M selbst frisst, so frisst sich M nach Definition nicht, denn M frisst genau die Monster, die sich selbst fressen.

② Falls sich M nicht selbst frisst, so frisst sich M nach Definition, denn M frisst genau die Monster, die sich selbst fressen.

Monster

Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.

Frage. Gibt es ein Monster, das genau die Monster frisst,
die sich nicht selbst fressen?

Antwort. Nein!

Beweis.
Angenommen, es gäbe ein solches Monster M.
Nach dem Axiom frisst sich M entweder selbst oder nicht.

① Falls sich M selbst frisst, so frisst sich M nach Definition nicht, denn M frisst genau die Monster, die sich selbst fressen.

② Falls sich M nicht selbst frisst, so frisst sich M nach Definition, denn M frisst genau die Monster, die sich selbst fressen.

Beide Fälle ergeben einen Widerspruch.

Monster

Axiom.
Jedes Monster frisst sich entweder selbst oder es frisst sich nicht selbst.

Frage. Gibt es ein Monster, das genau die Monster frisst,
die sich nicht selbst fressen?

Antwort. Nein!

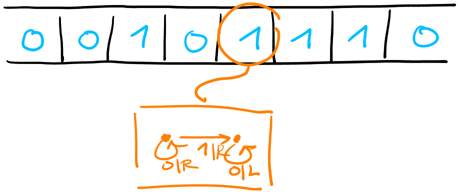
Beweis.
Angenommen, es gäbe ein solches Monster M.
Nach dem Axiom frisst sich M entweder selbst oder nicht.

① Falls sich M selbst frisst, so frisst sich M nach Definition nicht, denn M frisst genau die Monster, die sich selbst fressen.

② Falls sich M nicht selbst frisst, so frisst sich M nach Definition, denn M frisst genau die Monster, die sich selbst fressen.

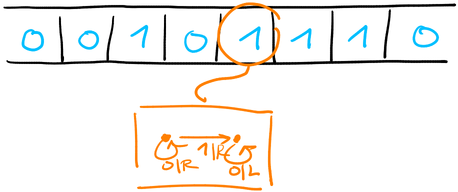
Beide Fälle ergeben einen Widerspruch.
Somit gibt es kein solches Monster. □

Unberechenbar! Turingmaschinen



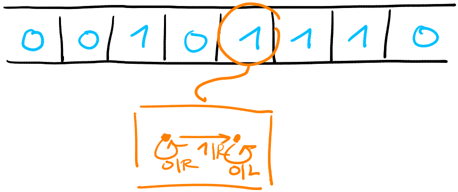
- Turingmaschinen sind ein einfaches, abstraktes Maschinenmodell, das alle gängigen Berechnungen und Algorithmen modelliert.

Unberechenbar! Turingmaschinen



- Turingmaschinen sind ein einfaches, abstraktes Maschinenmodell, das alle gängigen Berechnungen und Algorithmen modelliert.
- Turingmaschinen sind komplex genug, um Monster-artige Selbstbezüglichkeit zu ermöglichen.

Unberechenbar! Turingmaschinen

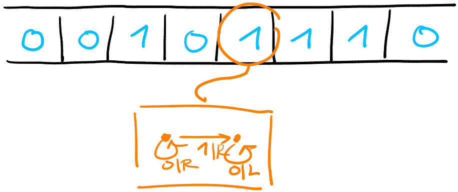


- Turingmaschinen sind ein einfaches, abstraktes Maschinenmodell, das alle gängigen Berechnungen und Algorithmen modelliert.
- Turingmaschinen sind komplex genug,
  um Monster-artige Selbstbezüglichkeit zu ermöglichen.

  Insbesondere:

  Das Halteproblem für Turingmaschinen ist *nicht* lösbar, d.h.:
  Es gibt *keine* Turingmaschine, die folgendes Problem löst:
  - Gegeben eine Turingmaschine T und einen Input x,
  - entscheide, ob T bei Input x terminiert oder nicht.

Unberechenbar! Turingmaschinen



- Turingmaschinen sind ein einfaches, abstraktes Maschinenmodell, das alle gängigen Berechnungen und Algorithmen modelliert.
- Turingmaschinen sind komplex genug, um Monster-artige Selbstbezüglichkeit zu ermöglichen.

  Insbesondere:

  Das Halteproblem für Turingmaschinen ist *nicht* lösbar, d.h.:
  Es gibt *keine* Turingmaschine, die folgendes Problem löst:
  - Gegeben eine Turingmaschine T und einen Input x,
  - entscheide, ob T bei Input x terminiert oder nicht.

  Denn: Sonst gäbe es eine Turingmaschine, die genau auf den Eingaben terminiert, die nicht auf sich selbst terminieren.

Unberechenbarkeit in der Praxis

Heutige Computer und Programmiersprachen
sind *nicht* ausdrucksmächtiger als Turingmaschinen!

Unberechenbarkeit in der Praxis

Heutige Computer und Programmiersprachen
sind *nicht* ausdrucksmächtiger als Turingmaschinen!

PS: Auch AI und Quantencomputer nicht.

Unberechenbarkeit in der Praxis

Heutige Computer und Programmiersprachen
sind *nicht* ausdrucksmächtiger als Turingmaschinen!
PS: Auch AI und Quantencomputer nicht.

Inbsesondere kann es *kein* Computerprogramm geben,
das folgendes Problem löst:

- Gegeben ein Programm P in Python und einen ganzzahligen Input x,
- entscheide, ob das das Programm P bei Input x das Ergebnis 2025 ausgibt oder nicht.

Unberechenbarkeit in der Praxis

Heutige Computer und Programmiersprachen
sind *nicht* ausdrucksmächtiger als Turingmaschinen!
PS: Auch AI und Quantencomputer nicht.

Inbsesondere kann es *kein* Computerprogramm geben,
das folgendes Problem löst:

- Gegeben ein Programm P in Python und einen ganzzahligen Input x,
- entscheide, ob das das Programm P bei Input x das Ergebnis 2025 ausgibt oder nicht.

Warum ist das relevant?

- Allgemeine, vollständige algorithmische Korrektheitsprüfung von Programmen ist z.B. in Python, C, Java, Haskell, LaTeX, ... *nicht* möglich.
  Einzelne Aspekte für konkrete Programme können individuell manchmal behandelt werden.

Unberechenbarkeit in der Praxis

Heutige Computer und Programmiersprachen
sind *nicht* ausdrucksmächtiger als Turingmaschinen!
PS: Auch AI und Quantencomputer nicht.

Inbsesondere kann es *kein* Computerprogramm geben,
das folgendes Problem löst:

- Gegeben ein Programm P in Python und einen ganzzahligen Input x,
- entscheide, ob das das Programm P bei Input x das Ergebnis 2025 ausgibt oder nicht.

Warum ist das relevant?

- Allgemeine, vollständige algorithmische Korrektheitsprüfung von Programmen ist z.B. in Python, C, Java, Haskell, LaTeX, … *nicht* möglich.
  Einzelne Aspekte für konkrete Programme können individuell manchmal behandelt werden.
- Terminierungsgarantie erfordert Einschränkung der Ausdrucksfähigkeit (z.B. Epigram).

Mehr Unberechenbarkeit/Unbeweisbarkeit

- Es gibt *keinen* Algorithmus, der entscheidet, welche polynomialen Gleichungen mit ganzzahligen Koeffizienten eine ganzzahlige Lösung besitzen.

$$x^2 - 1 = 0, \quad x^2 + 1 = 0, \quad x^{2025} + y^{2025} = z^{2025}$$

Mehr Unberechenbarkeit/Unbeweisbarkeit

- Es gibt *keinen* Algorithmus, der entscheidet, welche polynomialen Gleichungen mit ganzzahligen Koeffizienten eine ganzzahlige Lösung besitzen.

$$x^2 - 1 = 0, \quad x^2 + 1 = 0, \quad x^{2025} + y^{2025} = z^{2025}$$

- Für die meisten reellen Zahlen gibt es *keinen* Algorithmus, der ihre Dezimaldarstellung Schritt für Schritt ausgibt.

$$0{,}010010001\ldots, \quad 0{,}130238471985986298571273 61\ldots???$$

Mehr Unberechenbarkeit/Unbeweisbarkeit

- Es gibt *keinen* Algorithmus, der entscheidet, welche polynomialen Gleichungen mit ganzzahligen Koeffizienten eine ganzzahlige Lösung besitzen.

$$x^2 - 1 = 0, \quad x^2 + 1 = 0, \quad x^{2025} + y^{2025} = z^{2025}$$

- Für die meisten reellen Zahlen gibt es *keinen* Algorithmus, der ihre Dezimaldarstellung Schritt für Schritt ausgibt.

$$0{,}010010001\dots, \quad 0{,}13023847198598629857127361\dots???$$

- Man kann innerhalb der Mengenlehre *nicht* beweisen, dass die Mengenlehre widerspruchsfrei ist.

$$\emptyset, \quad \{\emptyset\}, \quad \{\emptyset, \{\emptyset\}\}, \quad \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}, \quad \dots$$

- ...

Epilog: Proofs $=$ Programs

Formalisierung!

Es ist möglich,

- mathematische Axiome
- Definitionen
- Sätze
- Beweise
- Beispiele

in geeigneten Programmiersprachen (Beweisassistenten) zu formalisieren.

Formalisierung!

Es ist möglich,

- mathematische Axiome
- Definitionen
- Sätze
- Beweise
- Beispiele

in geeigneten Programmiersprachen (Beweisassistenten) zu formalisieren.

Zum Beispiel: Automath [de Bruijn, 1967], HOL, Isabelle, Coq, Lean, Agda

Formalisierung!

Es ist möglich,

- mathematische Axiome
- Definitionen
- Sätze
- Beweise
- Beispiele

in geeigneten Programmiersprachen (Beweisassistenten) zu formalisieren.
Zum Beispiel: Automath [de Bruijn, 1967], HOL, Isabelle, Coq, Lean, Agda

- Es ist algorithmisch überprüfbar, ob solche Beweise korrekt sind.
- Es ist *nicht* algorithmisch überprüfbar, ob Sätze korrekt sind.

Quellen

- D. Hilbert. Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Kongreß zu Paris 1900. In: Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse aus dem Jahre 1900. Commissionsverlag der Dieterich'schen Universitätsbuchhandlung Lüder Horstmann, Göttingen, S. 253—297, 1900.
- K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte für Mathematik und Physik, 38, 173–198, 1931.
- A.M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society. s2–42(1), 230–265, 1937.
- S. 12: https://commons.wikimedia.org/wiki/File:P._Oxy._I_29.jpg
- S. 12: F. Waldhausen. The algebraic K-theory of spaces. In: Ranicki, A., Levitt, N., Quinn, F. (eds) Algebraic and Geometric Topology. Lecture Notes in Mathematics, 1126. Springer.
- S. 21: https://opc.mfo.de/detail?photo_id=541
- alle weiteren Photographien sind im public domain.