Grenzen der Mechanisierung

Prädikatenlogik

- Prädikatenlogik: Logik von Aussagen mit Prädikaten „ist rot", „ist eine Zahl" … F( ), G( ) …
- Quantoren: „alle" $\forall$, „einige" $\exists$
- Beispiel: Alle Katzen sind Säugertiere: $(\forall x)(\text{Katze}(x) \supset \text{Säugetier}(x))$
- Axiome wie: $\forall x\, F(x) \supset F(a)$: was für alle gilt, gilt für den Einzelfall
- Logische Wahrheiten wie: $\exists x(F(x) \vee G(x)) \supset \exists x F(x) \vee \exists x G(x)$
- Alle logischen Wahrheiten *sind* algorithmisch/mechanisch herleitbar.
  (*Gödels Vollständigkeitstheorem*), also wie in der Aussagenlogik.
- Aber: Es gibt *kein* allgemeines algorithmisches/mechanisches Verfahren, von einem Satz festzustellen, ob es sich um eine logische Wahrheit handelt!
  (*Churchs Theorem*). Die Prädikatenlogik im Allgemeinen ist nicht entscheidbar, im Unterschied zur Aussagenlogik.
- Beispiel: Goldbachs Vermutung lässt sich formulieren, jedoch nicht entscheiden
  $\forall x(\text{Zahl}(x) \wedge \text{Gerade}(x) \supset \exists z \exists y(\text{Primzahl}(z) \wedge \text{Primzahl}(y) \wedge x = y + z)$

Negative Theoreme

- Neben *Churchs Theorem*:
- *Tarskis Theorem*: Der Begriff der Wahrheit ist nicht formal repräsentierbar (durch Theoreme „Wahr(A)“, die genau dann beweisbar sind, wenn A wahr ist).
- *Gödels Erstes Unvollständigkeitstheorem*: Es gibt wahre Sätze, die nicht beweisbar sind (in konsistenten hinreichend ausdrucksstarken Systemen). Also dort gilt:
  Wahrheit ≠ Beweisbarkeit.
- *Gödels Zweites Unvollständigkeitstheorem*: Kein hinreichend ausdrucksstarkes System kann seine eigene Konsistenz beweisen.
- *Rices Theorem*: Es gibt kein allgemeines algorithmisches/mechanisches Verfahren, um eine nicht-triviale Eigenschaft eines Algorithmus zu entscheiden/erkennen.
  (Also z.B. ob zwei Algorithmen dasselbe berechnen, oder ob ein Algorithmus genau das tut, was er vermeintlich soll etc.)
- *Turings Theorem*: Es ist unentscheidbar, ob ein Programm irgendwann anhält.

Der Lügner

- (1) Satz (1) ist falsch.
- Angenommen: 1. Es gilt falsch = nicht-wahr
- Angenommen: 2. Wahrheitsprinzip: Satz „A“ ist wahr genau dann, wenn A
- Also: (1) ist entweder wahr oder nicht wahr.
  Ist (1) wahr, dann nach Wahrheitsprinzip: (1) ist falsch
  Ist (1) nicht wahr, dann nach Definition „falsch“ (1) falsch, also nach Wahrheitsprinzip ist (1) wahr.

  Ein Widerspruch!
- Lösung: Unterscheidung von Objektsprache und *Meta*sprache.
- (2) Satz (1) ist wahr

  gehört zur Metasprache. Kein Satz darf/soll über sich selbst reden.
  Tarskis Theorem reformuliert: „Wahrheit“ gehört nicht zur Objektsprache.

Russells Antinomie

- Für Eigenschaften sollte gelten: Zu jeder Eigenschaft gibt es die Menge der Gegenstände, welche diese Eigenschaft haben.
- $\forall F\ \exists y\ \forall x\ (x \in y \equiv F(x))$ - Komprehensionsprinzip
- Gewählte Eigenschaft: nicht Element seiner selbst sein: $x \notin x$
- Also nach Komprehensionsprinzip: $\exists y\ \forall x\ (x \in y \equiv x \notin x)$
  Nennen wir diese Menge „r“, also: $\forall x\ (x \in r \equiv x \notin x)$
  Was ist nun mit r selbst? Es muss gelten: $r \in r \equiv r \notin r$

  Ein Widerspruch!
- Lösung: Unterscheidung von Stufen und/oder Aufgabe des Prinzips.
  (D.h. eine Menge enthält nur Elemente, die stufenmäßig unter ihr liegen.)

Beweisbarkeit

- Die Prädikatenlogik kann die Zahlentheorie ausdrücken.
  Beispiel: $\forall x(\mathrm{Zahl}(x) \supset \exists y(\mathrm{Zahl}(y) \wedge y > x)$ etc. z.B. die Peano-Axiome
- Formeln einer Sprache können in Zahlen codiert werden (‚Gödelnummerierung'). Damit können auch Beziehungen und Eigenschaften von Formeln als Zahlenbeziehungen codiert werden.
- Deswegen kann ein hinreichend ausdrucksstarkes System (Prädikatenlogik + Zahlentheorie) über die Beziehungen *seiner* Formeln zu einander reden!
- Zu diesen Beziehungen gehört ‚x beweist y': Beweisbarkeit kann in der Sprache ausgedrückt werden (nicht nur in der Metasprache wie „Wahrheit")!
- Formeln können also über *ihre eigene Beweisbarkeit* reden.

Der Gödel-Satz

- Der Gödelsatz sagt: Ich bin nicht beweisbar. $G \equiv \neg\text{Beweisbar}(G)$
- Entweder ist der Gödel-Satz beweisbar oder nicht-beweisbar.
  Ist der Gödel-Satz beweisbar gilt aufgrund seines Inhalts: er ist nicht beweisbar.
  Ein Widerspruch! Das System wäre nicht korrekt.
  Also ist der Gödel-Satz nicht beweisbar. Das sagt aber der Gödel-Satz.
  Also gibt es einen wahren Satz, der nicht beweisbar ist (sein kann)!
- Insbesondere: Es gibt wahre Sätze der Zahlentheorie/Mathematik, die sich nicht beweisen lassen! Gegen Leibniz‘ Ideal (*Gödels Erstes Unvollständigkeitstheorem*). Daraus ergibt sich direkt das *Zweite Unvollständigkeitstheorem*, denn Konsistenz beweisen hieße zu beweisen, dass G nicht beweissbar ist, was inkonsistent wäre!
- Das gilt um so mehr je stärker ein Formales System ist - d.h. unbehebbar gibt es immer entsprechende Sätze.
  (Es gibt auch sachhaltigere Sätze der Zahlentheorie als G, für die dies gilt.)

Universelle Turing-Maschinen

- Eine Turing-Maschine ist ein abstrakter Automat. Alles was sich intuitiv überhaupt berechnen lässt, lässt sich mit einer Turing-Maschine (TM) berechnen (*These von Church und Turing*).
- TMs und ihre Programme lassen sich codieren. Eine Universelle TM ist eine TM, die beliebige andere TMs simulieren kann bzw. deren Programme ausführt: ein universeller Computer.
- Prädikatenlogische Herleitungen lassen sich durch TMs simulieren und TM-Berechnungen lassen sich prädikatenlogisch herleiten. Die beiden formalen Vorgehensweisen sind äquivalent (*Turings Theorem*).

Das Halteproblem

- Bezüglich von TMs lässt sich fragen: Hält die Berechnung der TM auf ihrem Input irgendwann an? Das ist das *Halteproblem*.
- Es gibt *kein* allgemeines algorithmisches/mechanisches Verfahren, um das Halteproblem zu lösen.
- Denn: Angenommen eine UTM würde die Nicht-Halter erkennen können.
  Was passiert, wenn die UTM ihren eigenen Code testet?
  Wenn sie anhält, erkennt sie sich als Nicht-Halter – ein Widerspruch!
  Ist sie ein Nicht-Halter, müsste sie dies erkennen und anhalten – ein Widerspruch!
  Beide Möglichkeiten scheiden aus. Eine solche TM kann es nicht geben.
- Aufgrund der Äquivalenz von Prädikatenlogischen Theorien und TMs überträgt sich das Problem auf die Entscheidbarkeit der Prädikatenlogik. Ein solches Verfahren zur Entscheidung kann es nicht geben (*Churchs Theorem*).

Fazit

- Bei komplexeren logischen Problemen gibt es Grenzen der Lösung durch Algorithmen (Mechanismen):
- Nicht jedes klar formulierte (formale) Problem kann entschieden werden (entgegen Leibniz‘ „calculemus“-Ideal)
- Nicht jede Wahrheit ist beweisbar.
- Eigenschaften von Programmen/Algorithmen müssen jeweils im Einzelfall überprüft und bewiesen werden, es gibt keine allgemeine Verfahren.

- All dies gilt in der Standard-Logik/Mathematik. Diese Grenzen bestehen, solange Konsistenz gefordert wird. Diese (partiell) aufzugeben führt zu ‚parakonsistenten Logiken‘, einem Gegenstand der Forschung.