Hilbert's Programm:

Baue die Mathematik von Grund auf neu, ausgehend von Axiomen, die konsistent und widerspruchsfrei sein müssen. Die Axiome sollen außerdem vollständig sein, d.h. alles Wahre soll daraus formal abgeleitet werden können.

Gödel zerstörte diesen Traum mit seinem Unvollständigkeitssatz.

Mathematische Ausdrücke:

z.B.:

$$\forall q \in \mathbb{N}_0 \; \exists p \in \mathbb{N}_0 \; \forall x,y \in \mathbb{N}_0 \; (p > q \wedge (x,y > 1 \Rightarrow (x \cdot y + p \mid x \cdot y + p + 2)))$$

Alphabet: $\Sigma = \{\wedge, \vee, \neg, (,), \forall, \exists, x, R_1, \dots, R_k\}$

z.B. $R_1(x_i, x_j, x_e) =$ „wahr" $:\Leftrightarrow x_i + x_j = x_e$

„Additionsrelation"

genauso: Multiplikationsrelation

Def.: Ein Satz ist ein Ausdruck $w \in \Sigma^*$ der Form:

$$Q_1 x_1 Q_2 x_2 \dots Q_e x_e \, \phi$$

wobei $Q_1, \dots, Q_e \in \{\forall, \exists\}$ und die Formel ϕ ist rekursiv wie folgt definiert:

1) $\phi = R_j(x_{i_1}, x_{i_2}, \dots, x_{i_n})$ für $1 \le j \le k$, $1 \le i_n \le e$ ist Formel.

2) $\phi = \phi_1 \wedge \phi_2$ oder $\phi = \phi_1 \vee \phi_2$ oder $\phi = \neg \phi_1$

wobei ϕ_1, ϕ_2 kürzere Formeln sind.

Bsp:

$\forall x_1 \forall x_2 \left(R_3(x_1, x_2) \vee R_3(x_2, x_1) \right)$

Wobei $R_3(x_1, x_2) =$ „wahr" $:\Leftrightarrow x_1 \leq x_2$

Dieser Satz ist wahr.

Ersetzt man die $\leq$-Relation durch die $<$-Relation, so ist der Satz falsch.

Für eine fest vorgegebene Menge von Relationen $\{R_1, \ldots, R_k\}$ sei L die Sprache der wahren Sätze:

$L = \{Q_1 x_1 \ldots Q_e x_e \phi = s \mid S \text{ ist wahrer Satz}\}$

Satz (von Church)

L ist nicht rekursiv, falls unter den Relationen $R_1, \ldots, R_k$ die Additionsrelation und die Multiplikationsrelation ist.

Bsp: Es ist nicht entscheidbar, ob ein Polynom über mehreren Variablen eine ganzzahlige Nullstelle besitzt.

Der Gödelsche Unvollständigkeitssatz beruht auf den folgenden Annahmen über „Beweise" von Sätzen:

1) Ein Beweis π eines Satzes s kann von einer DTM überprüft werden, d.h.:

$$\{s\pi \mid \pi \text{ ist Beweis von } s\}$$

ist rekursiv.

2) Ist π Beweis von s, so ist s wahr.

Unter diesen Annahmen gilt:

<u>Satz:</u> Die Sprache L' aller beweisbaren Sätze ist rekursiv aufzählbar

$$L' := \{ s \mid \exists \text{ Beweis } \pi \text{ für } s\}$$

<u>Beweis:</u> Zu einem gegebenen Satz überprüft man alle möglichen „Beweise" der Reihe nach, nach aufsteigender Länge mit Hilfe der DTM aus Annahme 1). Gibt es also einen gültigen Beweis für s, so wird dieser in endlicher Zeit gefunden. □

Gödelscher Unvollständigkeitssatz

Es sei L die Sprache der wahren Sätze, die mittels Additions- und Multiplikationsrelation formuliert werden können. Dann gibt es Sätze in L, die nicht beweisbar sind.

Beweis: Widerspruchsbeweis: Nehme an, dass $L = L'$. Betrachte gegebenen Satz s. Dann ist entweder s oder seine Negierung wahr. Wende die DTM aus dem vorigen Beweis parallel an auf s und die Negierung von s. Da einer von beiden wahr ist, gibt es (unter der Annahme $L = L'$) einen Beweis für einen von beiden. Dieser Beweis wird in endlicher Zeit gefunden. $\Rightarrow$ L ist rekursiv $\lightning$ (Satz von Church). □

Komplexitätsklasse P und NP-Vollständigkeit

Def: Es sei M eine DTM auf dem Ein-

gesamtalphabet Σ. Die worst-case Rechenzeit $t_M(n)$ ist die maximale Anzahl Rechenschritte, die M auf einer Eingabe aus Σ^n durchführt.

Def.: P ist die Klasse der Sprachen (Entscheidungsprobleme), für die es eine DTM M gibt, die L entscheidet und

$$t_M(n) \leq p(n)$$

für ein Polynomfkt. p.

Eine Sprache (Problem) in P heißt **effizient entscheidbar**. Der zugehörige Algorithmus (DTM) wird dann **effizient** genannt.

Problem: Wie zeigt man, dass ein Problem nicht effizient lösbar ist?

i) Zeige, dass das Problem nicht entscheidbar ist! (Aber viel stärkere Aussage!)

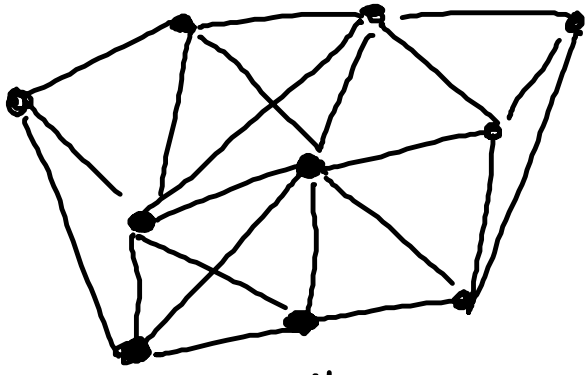
ii) Zeige, dass die erwartete Ausgabe nicht polynomial in der Eingabegröße beschränkt ist. (Normalerweise ist so ein Problem falsch gestellt.)

iii) Direkte Beweise nur in seltenen Fällen bekannt (z.B. Presburger Arithmetik) und schwer zu finden.

Keinem dieser 3 Ansätze ist praktisch zufriedenstellend.

Def.: Es sei $G=(V,E)$ ein ungerichteter Graph. Dann bildet $V' \subseteq V$ eine Clique, wenn für alle Paare $u,v \in V'$ mit $u \neq v$ eine Kante zwischen u und v existiert.

Bsp.:



V'

Wir betrachten das Cliquenproblem CLIQUE:

1. Variante: Gegeben $G=(V,E)$ und $k \in \mathbb{N}$. Gibt

eine Clique $V' \subseteq V$ mit $|V'| = k$?
(Entscheidungsproblem).

2. Variante: Gegeben $G = (V, E)$. Berechne das größte $k \in \mathbb{N}$, so dass G eine Clique der Größe k besitzt. (Optimierungsproblem)

3. Variante: Gegeben $G = (V, E)$. Berechne $V' \subseteq V$ in G, so dass V' Clique ist und $|V'|$ maximal. (Optimierungsproblem)

Satz: Gibt es für eine der 3 Varianten des Cliquenproblems einen effizienten Algorithmus (DTM), dann auch für die anderen beiden.

Bem: Beachte, dass es unterschiedliche Arten gibt, einen Graphen zu kodieren (z.B. Adjazenzmatrix, Adjazenzlisten etc.). Diese unterscheiden sich in der Größe jedoch nur um polynomielle Faktoren.

Beweis des Satzes: Kann man Variante 3 effizient lösen, dann auch Varianten 1 und 2. Ein effiz. Alg. für Variante 2 löst auch Variante 1.

③ → ② ⇒ ①

Zeige noch: Kann man Variante 1 effizient lösen, dann auch Variante 3.
→ nächste Woche ...