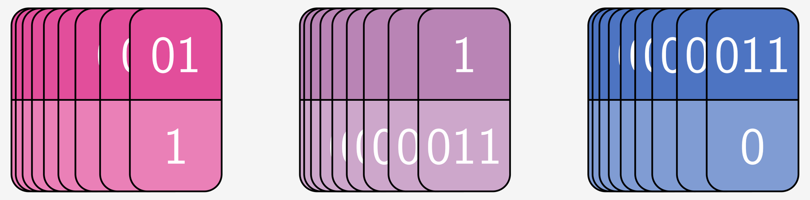
Domino spielen mit Emil Post

Leonie Kayser

MAX-PLANCK-INSTITUT FÜR MATHEMATIK IN DEN NATURWISSENSCHAFTEN

Das Postsche Korrespondenzproblem (PCP)

Das folgende Rätsel geht auf *Emil Post* zurück: Gegeben ist ein Satz beschrifteter Dominosteine, wobei von jeder Sorte beliebig viele Steine zur Verfügung stehen. Ist es möglich, die Steine so anzureihen, dass oben und unten die gleiche Zeichenfolge zu lesen ist? Ein Beispiel:



Diese gegebene *Instanz* des Rätsels ist lösbar, die kürzeste Lösung ist unten dargestellt. Aber wie kann man systematisch eine Lösung finden, oder *entscheiden*, dass es keine Lösung gibt? Aus mathematischer Perspektive stellt sich heraus, dass dies nicht möglich ist, zumindest nicht algorithmisch!

Emil Post (1897–1954)

Entscheidungsprobleme überall

In der Mathematik und Informatik fragen wir nach systematischen Lösungen zu vielen verschiedenen Problemen:

- Hat eine gegebene Gleichung eine Lösung, wenn ja, welchen Wert hat sie?
- Was ist die kürzeste Strecke von Ort A nach B? Oder doch eine Rundreise?
- Was ist die Mindestanzahl der Farben, um eine Landkarte zu färben?
- Was ist der optimale Zug in einem Spiel Tic-Tac-Toe? Oder in Schach?
- Ist eine gegebene Zahl p eine Primzahl, oder hat sie eine Zerlegung $p = a \cdot b$?

Ein solches Problem heißt *entscheidbar*, wenn ein geeignetes Computerprogramm mit genug Ressourcen die Frage beantworten kann. Eine weitergehende Frage ist, ob sich entscheidbare Probleme auch *effizient* lösen lassen.

Wie kann ein Problem *nicht* entscheidbar sein?!

Ein grundlegendes Problem der Informatik ist das Halteproblem:

- Gegeben ein Computerprogramm `"Prog"` und eine Eingabe `"Input"`
- Frage: Terminiert die Ausführung oder gibt es eine Endlosschleife?

Alan Turing bewies 1937, dass **kein** Algorithmus diese Frage in endlich vielen Schritten beantworten kann.

Angenommen, es gäbe ein hypothetisches Programm, welches bei Eingabe `"Prog"` und `"Input"` mit *ja/nein* antwortet. Dann gibt es auch ein Programm `F`, welches bei Eingabe `"Prog"`

1. *ja* ausgibt, wenn das Programm `Prog` auf die Eingabe `"Prog"` (seine eigene Beschreibung) **nicht** terminiert;
2. oder sich anderenfalls in eine Endlosschleife begibt.

Doch was würde passieren, wenn man `F` seine *eigene* Beschreibung `"F"` als Eingabe geben würde?

- Wenn die Ausgabe *ja* wäre, dann bedeutet dies, dass `F` bei Eingabe `"F"` **nicht** hält, dies ist also nicht möglich.
- Wenn `F` auf die Eingabe `"F"` in eine Endlosschleife gerät, dann erfüllt `F` nicht seinen Zweck!

Dieser Widerspruch zeigt, dass es das Programm `F` nicht geben kann, das Halteproblem also nicht algorithmisch gelöst werden kann.

Einen Computer mit Dominosteinen simulieren

Auf den ersten Blick hat das Dominospiel von Emil Post wenig mit dem Halteproblem zu tun. Doch wir können die Unentscheidbarkeit des Halteproblems auf das PCP übertragen, dies nennt man eine *Reduktion*. Die Idee ist die folgende: Angenommen wir haben Problemstellungen A, B, und eine effektive Möglichkeit, ein konkretes Problem aus A in ein Problem aus B zu übersetzen. Dann gilt

- Ist Problem B effektiv lösbar, so kann auch A effektiv gelöst werden, indem zu B übersetzt und dann gelöst wird.
- Ist umgekehrt A ein schwieriges Problem, etwa unentscheidbar, so muss dies auch auf B zutreffen.

Es stellt sich heraus, dass man die Verhaltensweise eines Computerprogramms (genauer: das Modell einer *Turingmaschine*) in einen Satz Dominosteine übersetzen kann, sodass das Programm genau dann terminiert, wenn das PCP-Rätsel der Dominosteine eine Lösung hat. Somit ist das Postsche Korrespondenzproblem ebenfalls unentscheidbar.

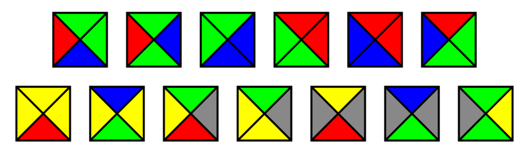
Unentscheidbare Probleme

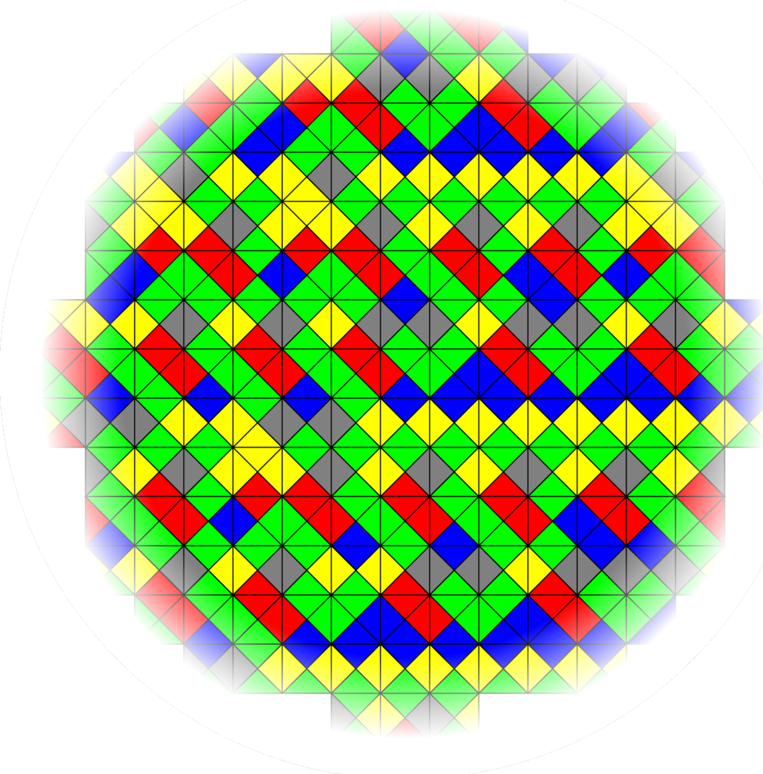
Folgende Fragen lassen sich nicht von einem Algorithmus in endlich vielen Schritten beantworten. Dies bedeutet nicht, dass nicht *konkrete* Fälle gelöst werden können, sondern dass das *allgemeine* Problem unentscheidbar ist.

- Das Halteproblem (und das Postsche Korrespondenzproblem) bilden die wohl grundlegendsten unentscheidbaren Probleme: Alle weiteren Probleme in dieser Liste sind unentscheidbar, weil sie das Halteproblem oder PCP im obigen Sinne „simulieren“ können.
- Matrix Mortality Problem: Gegeben sechs ganzzahlige 3×3-Matrizen $A_1, \ldots, A_6$, gibt es ein Produkt der Matrizen (mit Mehrfachverwendung), welches die Nullmatrix ergibt? Ein Beispiel mit zwei Matrizen:

$$\begin{pmatrix}0&1&-1\\0&1&-1\\1&0&1\end{pmatrix}\begin{pmatrix}-1&0&1\\-1&-1&0\\-1&1&1\end{pmatrix}^2\begin{pmatrix}0&1&-1\\0&1&-1\\1&0&1\end{pmatrix}^3\begin{pmatrix}-1&0&1\\-1&-1&0\\-1&1&1\end{pmatrix}^4\begin{pmatrix}0&1&-1\\0&1&-1\\1&0&1\end{pmatrix}^2\begin{pmatrix}-1&0&1\\-1&-1&0\\-1&1&1\end{pmatrix}\begin{pmatrix}0&1&-1\\0&1&-1\\1&0&1\end{pmatrix}\begin{pmatrix}-1&0&1\\-1&-1&0\\-1&1&1\end{pmatrix}^2\begin{pmatrix}0&1&-1\\0&1&-1\\1&0&1\end{pmatrix}=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}$$

- Satz von Richardson: Gegeben eine Funktion $f(x)$ verkettet aus ganzen Zahlen, x, $\sin(x)$, $+$, $-$, $\times$. Nimmt $f(x)$ nur positive Werte an?
- Hilberts 10. Problem: Hat eine Polynomgleichung $p(x_1, \ldots, x_n) = 0$ eine ganzzahlige Lösung?
- Kartenspiele: Welcher Spieler in einem *Magic: The Gathering* Spiel gewinnt, selbst wenn alle weiteren Züge erzwungen sind?
- Wang-Kacheln: Gegeben einen Satz farbiger quadratischer Kacheln, kann man damit die (unendliche) Ebene farblich passend parkettieren? Dabei sind von jedem Typ beliebig viele Kacheln vorhanden, aber die Kacheln dürfen nicht gedreht werden. Ein mögliches Beispiel, welches nur eine Parkettierung ohne Wiederholungen zulässt:





- Formale Sprachtheorie: Gegeben zwei kontextfreie Grammatiken beschreiben sie die gleiche Sprache? Dabei ist eine kontextfreie Grammatik eine Liste von Ersetzungsregeln, wie

  [Start] → [Subjekt][Verb][Objekt], [Objekt] → [Adjektiv][Objekt], [Objekt] → *Domino*, …

  und die davon gebildete Sprache sind alle Sätze, die nach diesen Regeln gebildet werden können.