MATHEMATIQ

Der Newsletter der MathSIG
(Interessensgruppe innerhalb der Mensa Österreich)

Ausgabe 27

http://www.hugi.scene.org/adok/mensa/mathsig/

Editorial

Liebe Leserinnen und Leser!

Dies ist die siebenundzwanzigste Ausgabe von MATHEMATIQ, dem Newsletter der MathSIG. Die MathSIG wurde gegründet, um die spezifischen Interessen mathematisch hochbegabter Menschen zu fördern. In erster Linie soll sie sich also den Themengebieten Mathematik, Informatik, Physik und Philosophie widmen. Beiträge von Lesern sind herzlich willkommen. Wenn in ihnen mathematische Sonderzeichen vorkommen, bitte ich aber, sie zwecks möglichst einfacher und fehlerfreier Formatierung im TEX-Format einzusenden. Als Vorlage ist eine Fassung des jeweils aktuellen Newsletters im TEX-Format auf Anfrage bei mir erhältlich. Außer Artikeln sind natürlich auch Illustrationen für das Titelblatt willkommen. Die Rechte an diesen müssen aber eindeutig bei euch selbst liegen, Kopieren von Bildern aus dem Internet ist nicht erlaubt.

Hinweis: Autoren sind für den Inhalt ihrer Artikel oder Werke selbst verantwortlich. Die in MATHEMATIQ veröffentlichten Beiträge widerspiegeln ausschließlich die Meinung ihrer Autoren und nicht jene des Vereins Mensa. Die Zusendung von Beiträgen gilt auch als Einverständnis zu deren Veröffentlichung in MATHEMATIQ.

Diese Ausgabe beschäftigt sich erneut mit dem P-NP-Problem.

In diesem Sinne: Viel Spaß beim Lesen und Lernen!

Claus D. Volko, cdvolko@gmail.com

Überlegungen zum P-NP-Problem

Wer die einschlägigen Artikel in den vergangenen Ausgaben noch nicht gelesen hat, möge zuerst diese lesen.

Alle Probleme in *P* haben die Gemeinsamkeit, dass der Suchraum polynomiell ist, wenn man so vorgeht, dass man alle relevanten Lösungen durchprobiert und mit einem polynomiellen Algorithmus (der ja existiert, weil *P* eine Teilmenge von *NP* ist) überprüft. Das bedeutet aber auch, dass der Suchraum maximal cn^k verschiedene Lösungen enthalten kann, wenn n die Größe der Eingabedaten ist und k und c Konstanten. Sonst wäre er nicht polynomiell. Die Größe der Datenstruktur, die genügt, um diesen Suchraum zu repräsentieren, beträgt daher $k \log n + \log c$ bit.

Ein jedes Problem in *P* erlaubt also, dass alle relevanten Lösungen in einer Anzahl von Bit dargestellt werden können, die sich zur Größe der Eingabedaten logarithmisch verhält, also viel kleiner als die Eingabedaten ist.

Ein Beispiel dafür: Das Problem, ob zwei Zahlen a und b einander teilerfremd sind, ist in *P*, denn ein gemeinsamer Teiler der beiden Zahlen muss zwischen 1 und a liegen, wenn a die kleinere der beiden Zahlen ist. Man muss also nur a verschiedene Lösungen durchprobieren. Das Problem ist polynomiell. Die Zahl a wird durch $1 + \log a$ bit dargestellt.

Ein etwas komplexeres Beispiel: Das Problem 2-COL, das sich damit beschäftigt, ob ein Graph mit nur zwei Farben färbbar ist, so dass zwei miteinander verbundene Knoten zwei verschiedene Farben zugewiesen bekommen, ist in *P*. Die Datenstruktur der Lösung scheint zunächst so beschaffen zu sein, dass es 2^n verschiedene Zahlenwerte gibt, wobei n die Anzahl der Knoten ist. Schließlich liegt es nahe, dass man eine Datenstruktur aus n bit verwendet, wobei jedes Bit entweder 0 oder 1 sein kann, je nachdem, welche Farbe dem Knoten zugewiesen wurde. Tatsächlich aber lässt sich jede Lösung eindeutig darstellen, indem ich einfach nur die Nummer eines Knoten abspeichere, der eine bestimmte Farbe zugewiesen bekommt - denn alle anderen Farbwerte können von einem Algorithmus in polynomieller Zeit ermittelt werden. Es geht sogar noch einfacher: Ich kann auch einen Anfangsknoten festlegen und einfach die Farbe abspeichern, die er zugewiesen bekommt, also 0 oder 1. Eine Instanz dieses Problems, die nicht mit zwei Farben färbbar ist, hat als Lösungsmenge die leere Menge.

Das heißt: Wäre $P = NP$, könnte ich dies beweisen, indem ich für ein *NP*-vollständiges Problem eine Darstellung der Lösungen fände, deren Größe $k \log n + \log c$ bit beträgt. Nehme ich ein *NP*-vollständiges Problem, zum Beispiel *SAT*, her, müsste ich zeigen können, dass ich jede Lösung so komprimieren kann, dass sich die Größe der komprimierten Lösung im Vergleich zur Originallösung asymptotisch betrachtet logarithmisch verhält. Das ist nur dann möglich, wenn ich bei einer Anzahl von n zu ermittelnden Werten bereits durch die Ermittlung von $\log n$ Werten alle anderen Werte eindeutig bestimmen kann. Wenn ich also zwei Variablen in *SAT* habe, muss die Angabe des Wertes einer Variablen genügen, um den Wert der zweiten Variablen eindeutig zu bestimmen; bei drei bis vier Variablen müssen zwei Werte genügen, bei fünf bis acht Variablen drei Werte usw. Wenn ich für eine einzige konkrete Instanz von SAT beweisen kann, dass sie auf diese Weise nicht lösbar ist, dann ist damit $P \neq NP$ bewiesen. Jedenfalls zumindest sofern meine hier dargelegte "line of reasoning" stimmt.

Claus D. Volko, cdvolko@gmail.com