MATHEMATIQ

Der Newsletter der MathSIG
(Interessensgruppe innerhalb der Mensa Österreich)

Ausgabe 21

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

Editorial

Liebe Leserinnen und Leser!

Dies ist die einundzwanzigste 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 wieder mit dem *P*-*NP*-Problem.

In diesem Sinne: Viel Spaß beim Lesen und Lernen!

Claus D. Volko, cdvolko@gmail.com

Könnte man das P-NP-Problem so lösen?

Als ich vor einigen Tagen meinen dreißigsten Geburtstag feierte, wünschte mir ein Gratulant, mir möge es in diesem neuen Lebensjahr endlich gelingen, das *P-NP*-Problem (siehe frühere Ausgaben von MATHEMATIQ) zu lösen. Daraufhin war ich wieder angestachelt und habe mich bis Mitternacht mit dem Problem beschäftigt, obwohl ich am nächsten Tag wieder arbeiten musste. Dabei kam die folgende Überlegung heraus:

Ein Problem ist genau dann in *NP*, wenn es einen polynomiellen Algorithmus (in Bezug auf die Größe der Eingabedaten) gibt, der überprüft, ob eine gegebene Lösung korrekt ist.

Ein Problem ist genau dann in *P*, wenn es in *NP* ist und zusätzlich die Anzahl der Lösungen, die überprüft werden müssen, so dass mit hundertprozentiger Sicherheit die korrekte Lösung darunter ist, polynomiell ist (ebenfalls in Bezug auf die Größe der Eingabedaten).

Ich definiere:

Lösungsraum: ist die Anzahl der Lösungen, die es überhaupt geben kann, egal ob korrekt oder nicht.

Suchraum: ist die Anzahl der Lösungen, die unbedingt durchsucht werden müssen, damit mit hundertprozentiger Sicherheit die korrekte Lösung darunter ist (ohne Raten und ohne vorherige Kenntnis der korrekten Lösung).

Die Frage ist, ob es Probleme gibt, die in *NP*, aber nicht in *P* liegen.

Bekannt ist: Es gibt *NP*-vollständige Probleme; nur wenn ein *NP*-vollständiges Problem in *NP*, aber nicht in *P* liegt, gibt es Probleme, die in *NP*, aber nicht in *P* liegen.

Ein *NP*-vollständiges Problem ist k-*SAT*: Gegeben sind n boolesche Variablen (können die Werte wahr oder falsch annehmen), diese kommen in c Clauses (Disjunktionen) vor, jeweils k Variablen pro Clause; alle Clauses sind miteinander durch Konjunktion überprüft.

Ich möchte zeigen, dass der Suchraum bei k-*SAT* nicht polynomiell und daher k-*SAT* nicht in *P* enthalten ist.

Der Lösungsraum beträgt 2^n, da jede Variable entweder wahr oder falsch sein kann, also genau zwei Werte annehmen kann. Der tatsächliche Suchraum ist kleiner als 2^n, aber im allgemeinen Fall nicht polynomiell. Der Grund: Es gibt keine andere Möglichkeit vorzugehen, als einer Variablen einen Wert zuzuweisen und dann zu schauen, wie sich das auf die anderen Variablen auswirkt. Wenn k-*SAT* in *P* wäre, dann müsste es in jeder Formel die gleiche Anzahl von Variablen geben, deren Wert

man festlegen muss, so dass sich die Werte aller anderen Variablen ergeben. Wenn das t Variablen sind, dann ist der Suchraum $\binom{n}{t}$. Wenn t konstant ist, ist das polynomiell. t ist aber nicht konstant. Vielmehr stellt die Anzahl der Clauses eine obere Schranke für t dar. t kann also unter Umständen so groß sein wie die Anzahl der Clauses. Da die Anzahl der Clauses bei k-*SAT* aber variabel ist, ist der Suchraum nicht polynomiell (bei der Angabe einer oberen Schranke käme n im Exponent vor).

Bei der Diskussion darüber im österreichischen Informatik-Forum zeigte sich, dass die Schwachstelle an diesen Ausführungen in der Aussage "Es gibt keine andere Möglichkeit vorzugehen, als einer Variablen einen Wert zuzuweisen und dann zu schauen, wie sich das auf die anderen Variablen auswirkt" besteht. Diese Aussage ist unbewiesen. Offenbar ist das aber auch die einzige Schwachstelle - zumindest konnte keine weitere gefunden werden. Dass ein Problem, das in *NP*, aber nicht in *P* liegt, einen nicht-polynomiellen Suchraum haben muss, lässt sich leicht beweisen; es ist eine logische Folgerung aus obigen Definitionen. Die Schwierigkeit besteht bloß darin zu zeigen, dass es nicht möglich ist, den Suchraum zu generieren, indem man einer Anzahl von Variablen einen Wert zuweist, wobei diese Anzahl der Variablen höchstens polynomiell in Bezug auf die Gesamtzahl der Variablen ist, sondern dass dieses Verhältnis nicht-polynomiell ist. Einige Kollegen meinten, ich wäre der Lösung des Problems nicht nähergekommen, sondern hätte es nur auf eine andere Ebene verlagert. Wie dem auch sei, vielleicht ist das Problem auf dieser Ebene leichter lösbar.

In diesem Zusammenhang möchte ich vorschlagen, vielleicht gar nicht über *SAT* zu diskutieren, sondern über ein anderes, anschaulicheres Problem:

Gegeben ist eine Konstante c. Gesucht sind n natürliche Zahlen ungleich 1 (denn 1 ist das neutrale Element der Multiplikation, was das Problem trivial machen würde), die miteinander multipliziert diese Konstante c ergeben.

Dieses Problem scheint mir genauso komplex wie *SAT* zu sein, es scheint mir aber einfacher zu analysieren zu sein. Ich wüsste für dieses Problem auch keine andere Möglichkeit, als nach einem Algorithmus vorzugehen, der für n - 1 Variablen die Werte durchiteriert, wobei sich der Wert der n-ten Zahl ergeben würde. Vielleicht könnte man für dieses Problem leichter als für *SAT* zeigen, dass es gar nicht anders gelöst werden kann. Vorschläge dazu wären sehr willkommen.

Claus D. Volko, cdvolko@gmail.com