

# DER MANN MIT DEM ROBOTER UND DEM ZAUBERWÜRFEL

## Professor Dr. Peer Johannsen lehrt Informatik im Studiengang Mechatronik



>> „Könnten Sie bei der Gelegenheit den Roboter vorführen, der den Zauberwürfel lösen kann?“

*Dies ist eine der häufigsten Fragen, die ich seit meinem Start an der Hochschule Pforzheim im Oktober 2010 gehört habe. Ich hatte das Glück, meine Arbeit als Professor für Informatik und Software-Engineering gleichzeitig mit dem ersten Jahrgang Mechatronik-Studierender zu beginnen. So bot sich die einmalige Gelegenheit, einen neuen Studiengang von Anfang mitzugestalten und aufzubauen.*

Mechatronik, – ein Zusammenschluss der drei Gebiete Informatik, Mechanik und Elektrotechnik – ist eine junge Disziplin. Viele technische Produkte bestehen heutzutage aus komplexen mechanischen Komponenten, die durch eine aufwändige, intelligente Elektronik gesteuert werden. In einem modernen Automobil finden sich neben der umfangreichen Mechanik und Motortechnik mittlerweile bis zu 70 spezielle Prozessoren und Mikrocontroller. Diese steuern eingebettete Teilsysteme wie zum Beispiel die Einspritzregelung, das ABS, das ESP, den Tempomat oder die Zentralverriegelung. Auch die Regelung einer ganz normalen Waschmaschine erfolgt heute nicht mehr rein mechanisch und elektrisch. Stattdessen wird die Steuerung von einem Mikrochip übernommen, der über mehr Rechenleistung verfügt als ein Bürocomputer vor zehn Jahren.

Mechatronische Systeme sind Systeme, in denen eine komplexe mechanische Konstruktion durch Computertechnologie gesteuert wird; es werden Informationen verarbeitet, die mit Hilfe von elektronischen Sensoren aus der Systemumgebung kommen. Beispiele sind Geschirrspülmaschinen, Kaffeevollautomaten, Industrieroboter und Fertigungsstraßen, Werkzeugmaschinen, Automaten und autonome intelligente Roboter. Mechatronik ist Teil vieler Alltagsanwendungen, ohne dass diese bislang explizit als mechatronisch bezeichnet werden. Heutzutage findet die Herstellung solcher Produkte weltweit und in Zusammenarbeit vieler Unternehmen statt. Um mechatronische Systeme zu entwickeln, zu optimieren und einzusetzen, ist interdisziplinäres Wissen über Mechanik, Elektrotechnik und Informatik sowie systembezogene Methodik gefragt.

Aufgabe der Informatik innerhalb der Mechatronik ist es, die notwendigen Kenntnisse über die Programmierung und den Entwurf von Software für eingebettete Systeme und Mikrocontroller zu vermitteln und den Aufbau und die Funktionsweise von Computern und digitaler Hardware zu erklären.

Die Informatik-Komponente von technischen Systemen ist für Studierende zunächst oft schwer fassbar, da man die Funktionsweise von Software und die Vorgänge innerhalb eines Mikroprozessors nicht direkt sehen kann. Zum Studienbeginn war es mir wichtig, die Programmierung und Steuerung eines kleinen mechatronischen Systems einmal anschaulich anhand eines konkreten Problems zu erläutern. Daher habe ich in den Informatik-Veranstaltungen des ersten und zweiten Semesters gemeinsam mit den Studierenden einen kleinen intelligenten Roboter gebaut, der in der Lage ist, selbständig einen beliebig verdrehten Zauberwürfel zu lösen und in seine Ausgangsstellung zurückzudrehen.

Die verschiedenen Teilgebiete der Mechatronik finden sich hier in den folgenden Aspekten:

- Die mechanische Konstruktion des Roboters muss in der Lage sein, alle notwendigen Drehungen des Würfels auszuführen (Mechanik).
- Elektromotoren sollen die Mechanik antreiben, und elektronische Sensoren sollen das Ausgangsproblem erkennen können, also die Farbverteilung der Quadrate auf allen Oberflächen des Würfels (Elektrotechnik).
- Ein Programm soll die Steuerung des Roboters übernehmen. Es soll das Einlesen der Farbinformationen des Würfels durchführen, die gelesenen Daten verarbeiten, daraus eine Folge von Lösungszügen berechnen und diese anschließend durch entsprechende Ansteuerung der Mechanik vom Roboter ausführen lassen (Informatik).

Das Verfahren, das den Roboter steuert, besteht aus einer Folge von einzelnen nacheinander ablaufenden Anweisungen und Tests. Die Resultate der Tests beeinflussen den Ablauf der Anweisungen. Solch ein Schema wird in der Informatik als Algorithmus bezeichnet. Algorithmen lassen sich mit Hilfe von Programmiersprachen in einer Form beschreiben, die für eine Maschine verständlich ist. Aus einem Algorithmus entsteht auf diese Weise ein Programm, das von einem Computer ausgeführt werden kann.

Das Beispiel des zauberwürfellosenden Roboters zeigt hierbei zwei wesentliche Aspekte des Entwurfs von Programmen in der Informatik:

- Ein allgemeines Verfahren zum Lösen des Zauberwürfels muss entworfen werden (Konstruktion eines Algorithmus).
- Dieses Verfahren muss in einer konkreten Programmiersprache formuliert und dabei an die Bedingungen eines konkreten Hardware-Systems angepasst werden (Umsetzen des Algorithmus in ein Programm für den konstruierten Roboter).

Der Roboter hat den Praxistest in der Vorlesung mit Bravour bestanden und ist zu einem beliebten Vorführobjekt auf dem Studieninformationstag, dem Girl's Day und bei der Langen Nacht der Wissenschaft geworden. Er veranschaulicht die Grundprinzipien der Mechatronik und verkörpert so den Studiengang Mechatronik an der Hochschule Pforzheim. Den Studierenden dient er als einführendes Beispiel, bevor diese im Rahmen von Projektarbeiten eigene Roboter entwerfen und programmieren. Als Nebeneffekt bin ich durch ihn mittlerweile inoffiziell als „der Mann mit dem Roboter und dem Zauberwürfel“ bekannt geworden. Dabei hat mich mein Berufsweg innerhalb der Informatik erst relativ spät in dieses spannende Teilgebiet der Robotik geführt.

Ich habe Informatik an der Christian-Albrechts-Universität zu Kiel studiert. Meine persönlichen Interessen innerhalb der Informatik liegen sowohl in der theoretischen Informatik (dem Entwurf von Algorithmen und Datenstrukturen), als auch in der praktischen und technischen Informatik (der Programmierung und Software-Entwicklung, also der Umsetzung von Algorithmen für konkrete Hardware-Systeme). Einer meiner Forschungsschwerpunkte ist die formale Verifikation digitaler Schaltkreise. Hier geht es darum, mit Hilfe von Verfahren der theoretischen Informatik (insbesondere der Automatentheorie) funktionale Fehler in den Entwurfsplänen digitaler Hardware zu finden, bevor diese in die Produktion gehen.

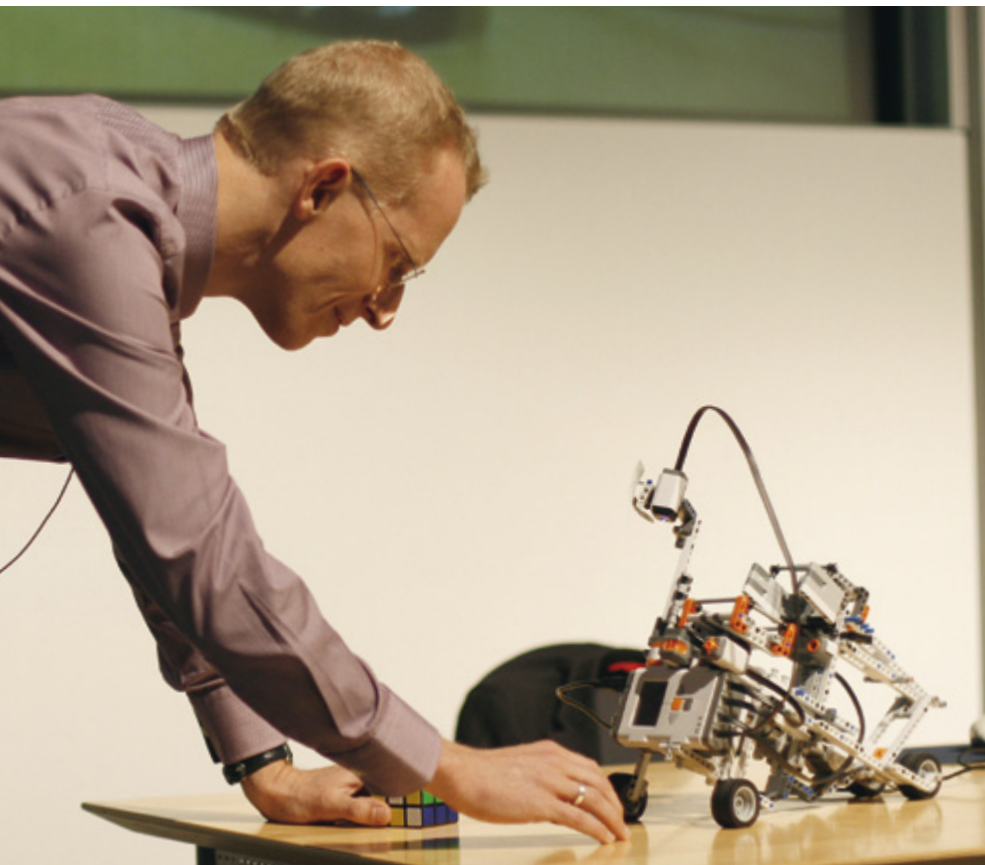
Im Anschluss an mein Studium habe ich in der Forschungsabteilung von Siemens in München mit der Arbeit an meiner Dissertation auf dem Gebiet der formalen Hardware-Verifikation begonnen. Im Rahmen meiner Promotion habe ich zwei Jahre in den USA in San Jose in Kalifornien im Silicon Valley gelebt und dort in der Chip-Entwicklung von Infineon

Technologies an meinem Promotionsthema geforscht. Einen längeren Auslandsaufenthalt kann ich allen Studierenden nur empfehlen. Neben den menschlichen und kulturellen Erfahrungen hat mich in dieser Zeit insbesondere die enge Verzahnung von wissenschaftlicher Forschung mit praktischer industrieller Anwendung geprägt und die Wahl meines weiteren Berufsweges mitbestimmt. Nach Abschluss der Promotion habe ich im Bereich Forschung und Entwicklung bei Siemens in München auf dem Gebiet der formalen Hardware-Verifikation weitergearbeitet. Kern meiner Tätigkeit war die Entwicklung von Inhouse-Software zur Schaltkreisverifikation, die in der Chip-Produktion bei Siemens und Infineon zum Einsatz kam.

Nach einem Wechsel zu Infineon erfolgte zwei Jahre später die erfolgreiche Ausgründung der Forschungsabteilung in die unabhängige Firma OneSpin Solutions GmbH. Dort habe ich bis zu meinem Wechsel nach Pforzheim den Aufbau eines Start-Ups zu einem eigenständigen Unternehmen mitgestaltet, das kommerzielle industrielle Verifikationssoftware für den Weltmarkt entwickelt und vertreibt. Meine langjährige Arbeit in einem Entwicklerteam von 20-30 Wissenschaftlern aus mitunter bis zu 15 verschiedenen Nationen sowie Forschungsk Kooperationen mit Universitäten und die Betreuung von Praktikanten und Studierenden im Rahmen von Bachelor- und Diplomarbeiten haben den Grundstein für mein langfristiges Berufsziel gelegt, wieder in die akademische Welt zurückzukehren. Die Verifikationssoftware, an der ich fast 12 Jahre aktiv mitentwickelt habe, ist in zahlreichen industriellen Chip-Projekten zum Einsatz gekommen. Diese enge Anbindung an die industrielle Praxis hat meine Wahl bewusst auf eine Hochschule für angewandte Wissenschaften fallen lassen.

Die Hochschule Pforzheim hat mir mit der Ausschreibung der Professur für Informatik im Studiengang Mechatronik dabei die Möglichkeit geboten, Informatik in einem hochinteressanten Umfeld zu betreiben. Die Arbeit meiner letzten Berufsjahre war von der Entwicklung von Algorithmen und deren konkreter Umsetzung in Software zur Unterstützung der Entwicklung von Hardware geprägt. Der Studiengang Mechatronik und die sich ergebenden Synergien mit anderen Studiengängen der Hochschule Pforzheim, wie z.B. der Technischen Informatik und dem Master Embedded Systems, bieten mir die Möglichkeit, sowohl im Bereich der Software-Entwicklung als auch der >





Hardware-Entwicklung aktiv, interdisziplinär und über Bereiche hinweg weiter zu arbeiten.

Weitere Punkte, sich für Pforzheim zu entscheiden, waren der nahe Wildpark, den ich gerne für eine kleine Auszeit in der Mittagspause nutze, und die Natur des Schwarzwaldes. Ich bin passionierter Marathonläufer und bevorzuge Laufstrecken in freier Natur. Langstreckenläufe sind für mich der ideale Ausgleich zur Arbeit am Bildschirm und eine hervorragende Möglichkeit, den Kopf frei zu bekommen oder sich neue Konzepte zu überlegen. Herausforderung und wesentlicher Unterschied zu meinen bisherigen Jahren als Läufer ist jedoch die Tatsache, dass es in der Gegend um Pforzheim so gut wie keine ebenen Laufstrecken gibt.

Zu meinen privaten Interessen gehören weiterhin die Beschäftigung mit logischen Rätseln und Puzzeln sowie mit Zaubertricks, insbesondere mathematisch basierten Tricks und Kartentricks. Thematisch schließt sich hier der Kreis zum Zauberwürfel und zur Informatik. Das Lösen des Zauberwürfels und das Lösen logischer Rätsel basieren beide weder auf Zufall noch auf Zauberei, sondern auf strukturierten Handlungsvorschriften, eben auf Algorithmen, dem Kernelement der Informatik. Auch hinter den meisten Zaubertricks steht im Wesentlichen nichts weiter als ein (in der Regel gut getarnter) Algorithmus, der vom Vorführenden durchgeführt wird, um ein bestimmtes Ergebnis zu erzielen.

Die Kunst ist es, zu einem gegebenen Problem eben solch einen allgemeinen Lösungsalgorithmus zu finden oder zu entwerfen:

- Ein Verfahren, das zu einem beliebig verdrehten Würfel eine Folge von Zügen liefert, die diesen wieder in die gelöste Stellung drehen.
- Ein Verfahren, das zu einem beliebigen Sudoku-Rätsel Schritt für Schritt abgearbeitet werden kann und zu einer Lösung führt.
- Ein Verfahren, das zwangsläufig bei der vom Vorführenden vorherbestimmten Karte endet, egal welche vermeintlich freie Wahl der Zuschauer unterwegs trifft.

Oder um es einmal sachlicher auf der Ebene der Informatik zu formulieren:

- Ein Programm in einer Programmiersprache für einen Computer zu schreiben, das für eine bestimmte Kategorie von Aufgaben ein korrektes Resultat berechnet.

Genau dieser Teilbereich der Informatik ist es, der mich am meisten fasziniert und begeistert: die Struktur eines logischen Problems zu analysieren und einen allgemeinen, möglichst effizienten und intelligenten Lösungsalgorithmus zu entwerfen.

Mein Interesse für die Informatik wurde bereits frühzeitig in jungen Jahren geweckt. Aufgewachsen bin ich in einem kleinen Dorf in Nordfriesland kurz vor

der dänischen Grenze zu einer Zeit, als PC noch ein Fremdwort war. Die ersten Heimcomputer kamen gerade auf den Markt, so dass zum ersten Mal auch Privatleute die Möglichkeit hatten, einen programmierbaren kleinen Computer zu kaufen. Diese Systeme (z.B. Atari XL, Commodore 64) wurden damals von den meisten Käufern jedoch nur als Spielekonsolen benutzt.

Gegen den Willen (und das Wissen) meines Vaters, der befürchtete, ich würde zu einem reinen Spieler werden, gelang es mir, meine Mutter dazu zu überreden, mir die Erlaubnis zu geben, mir von meinem Taschengeld solch einen Heimcomputer zu kaufen. Natürlich habe ich dieses Gerät auch zum Spielen benutzt, aber von Anfang an interessierte mich, wie es dazu kommen konnte, dass bunte Punkte als Raumschiffe über den Bildschirm gesteuert werden konnten, und wieso es den vom Computer gesteuerten Gegnern schlussendlich immer gelang, mich zu besiegen.

Im Alter von 8 Jahren begann ich, die eingebaute Programmiersprache (BASIC) meines Heimcomputers auszuprobieren und die ersten eigenen Programme (und natürlich Spiele) zu schreiben. Meine Liebe zum Programmieren war entdeckt, und ich begegnete zum ersten Mal dem Begriff des Algorithmus. Als ich dann noch erfuhr, dass es so etwas wie Informatik gibt, das man studieren kann und womit sich Geld verdienen lässt, stand mein späterer Berufswunsch fest.

Um besser zu verstehen, wie Computer im Inneren funktionieren, brachte ich mir selber Assembler, die eigentliche Maschinensprache von Computern, bei und schrieb darin wenig später die erste Variante des bekannten Tetris-Spiels für die damalige Atari ST Computer Serie. Erst später im Studium sollte ich lernen, dass es auch höhere Programmiersprachen wie C und C++ gibt, und erkennen, wie viel einfacher das Schreiben von Algorithmen und Programmen mit solchen Sprachen hätte sein können.

Der Entwurf von Algorithmen entspricht dem Entwurf von systematischen Lösungsverfahren. Programmieren ist für mich daher eng verwandt mit dem systematischen Lösen von Rätseln und Puzzeln. Das Wort „Puzzle“ stammt aus dem Englischen und steht allgemein für Rätsel. Das eingedeutschte Verb „puzzeln“ steht gemäß seiner ursprünglichen Bedeutung also weniger für das zufällige Ausprobieren, um eine Lösung zu finden, als vielmehr für die strategische Suche und das systematische Vorgehen, um möglichst schnell und elegant eine Lösung für ein vorliegendes Problem zu finden.



Ein möglicher Versuch, ein Sudoku-Rätsel zu lösen, besteht zum Beispiel darin, zufällig beliebige Zahlen in sämtliche freie Felder einzutragen und anschließend festzustellen, ob diese eine Gesamtlösung ergeben. Da es nur begrenzt viele verschiedene solcher zufälligen Zahlenkombinationen gibt, reicht es, alle Möglichkeiten systematisch nacheinander durchzuprobieren, und schon ist ein erster Lösungsalgorithmus entstanden. Leider ist die Gesamtzahl an Kombinationen, die schlimmstenfalls durchprobiert werden müssten, so groß, dass ein derartiges Vorgehen in der Regel viel zu lange dauern würde und in der Realität nicht praktikabel ist.

Das Ziel ist hier somit, einen besseren Algorithmus zu entwerfen, der die in einem Sudoku vorhandene Information (die Menge der anfangs vorgegebenen Zahlen) und die sich aus dieser Information ergebenden Konsequenzen (die logischen Schlussfolgerungen) ausnutzt, um gezielt in möglichst wenigen Schritten eine Gesamtlösung zu konstruieren. Der Algorithmus soll dabei allgemein gültig sein, d.h. er soll zu einer Lösung führen, egal welches konkrete Sudoku als Ausgangsproblem genommen wird. Viele Typen von Rätseln und viele reale Probleme weisen eine ähnliche Grundstruktur auf.

Bei der Verifikation von digitaler Hardware geht es darum, zu überprüfen, ob ein vorliegender Schaltkreis für alle möglichen Eingabewerte den jeweils korrekten Ausgabewert berechnet. Ziel ist es, einen konkreten Eingabewert zu finden, für den der Schaltkreis einen falschen Ausgabewert berechnet, oder aber zu beweisen, dass es solch einen Eingabewert nicht gibt. Die Gesamtzahl an Möglichkeiten ist jedoch viel zu groß, um

diese alle einzeln überprüfen zu können. Genauso wie im Beispiel des Sudokus wird hier ein systematisches Verfahren gesucht, das in einzelnen klar definierten Schritten vorgeht und zwangsläufig nach einer endlichen, möglichst geringen Anzahl von Schritten zu einer Lösung führt. Der Algorithmus soll allgemein sein, d.h. das Verfahren soll für beliebige Schaltkreise funktionieren, und es soll möglichst intelligent die Struktur des jeweils vorliegenden Schaltkreises ausgenutzt werden, um die Berechnung einer Lösung zu beschleunigen.

Informatik ist unter anderem die Wissenschaft vom Entwurf von Algorithmen. Die Kunst ist es dabei, zu einer gegebenen Klasse von Problemen einen allgemeinen Lösungsalgorithmus zu finden oder zu entwerfen und in ein Programm umzusetzen. Für sehr viele Problemklassen ist es vergleichsweise einfach zu überprüfen, ob eine konkrete Antwort eine Lösung darstellt. Doch nur weil das Überprüfen einer Antwort einfach ist, bedeutet dieses leider noch nicht, dass damit auch das Auffinden einer konkreten Lösung einfach ist.

Wenn ein Sudoku komplett mit Zahlen gefüllt ist, dann lässt sich leicht testen, ob diese Zahlen alle Sudoku-Bedingungen erfüllen. Wenn eine mathematische Funktion als Polynom gegeben ist, dann lässt sich der Funktionswert für jede konkrete Zahl leicht ausrechnen und somit feststellen, ob diese Zahl eine Nullstelle der Funktion ist. Wenn konkrete Eingabewerte für einen Schaltkreis gegeben sind, dann kann der Schaltkreis mit diesen Werten simuliert und die Korrektheit der Ausgabewerte direkt überprüft werden. Wenn die Beschreibung einer Folge von Drehungen des Zauberwürfels vorliegt, dann können diese Drehungen direkt ausgeführt und somit festgestellt werden, ob der Würfel anschließend gelöst ist.

Das Finden der konkreten Zahlen für die einzelnen Sudoku-Felder, das Berechnen einer Nullstelle zu einem Polynom, die Bestimmung von Eingabewerten, für die sich ein Schaltkreisentwurf fehlerhaft verhält, sowie die Konstruktion einer Folge von Lösungszügen für einen beliebigen verdrehten Zauberwürfel, all dieses kann hingegen extrem schwierig und aufwändig sein.

Das Finden der konkreten Zahlen für die einzelnen Sudoku-Felder, das Berechnen einer Nullstelle zu einem Polynom, die Bestimmung von Eingabewerten, für die sich ein Schaltkreisentwurf fehlerhaft verhält, sowie die Konstruktion einer Folge von Lösungszügen für einen beliebigen verdrehten Zauberwürfel, all dieses kann hingegen extrem schwierig und aufwändig sein.

Eine der berühmtesten (und berüchtigsten) Fragestellungen der Informatik ist folgende: Lässt sich für jedes Problem, bei dem es leicht ist, eine vorhandene Antwort auf Korrektheit zu überprüfen, auch auf einfache Art und Weise eine Lösung berechnen? Die hier beschriebenen Probleme sind allesamt lösbar, nötigenfalls durch Ausprobieren aller Möglichkeiten. Je nach Lösungsalgorithmus wird jedoch ein bestimmter Aufwand benötigt. Es stellt sich die Frage, ob ein vorhandener Algorithmus bereits der Beste ist, oder ob vielleicht bislang nur noch kein Mensch auf ein einfacheres Lösungsverfahren für das jeweilige Problem gekommen ist. In diesem Sinne gleicht der Entwurf von Algorithmen dem systematischen Lösen von Rätseln und Puzzeln. Die Suche nach einem möglichst effizienten und intelligenten Lösungsverfahren erfordert ein hohes Maß an Kreativität, das viele Menschen in naturwissenschaftlichen Fächern wie Mathematik und Informatik vielleicht gar nicht vermuten.

Da man meiner Ansicht nach Informatik nicht allein durch Tafelanschrieb und Vorlesung vermitteln kann, möchte ich Ihnen hier zum Abschluss noch ein kleines Puzzle mit auf den Weg geben, das ich für die Informatik-Vorlesung im Studiengang Mechatronik erstellt habe, um die Vorgehensweise beim Entwurf von Algorithmen zu verdeutlichen. Es ist ein typischer Vertreter der in diesem Artikel geschilderten Klasse von Problemen.

Die 9 abgebildeten quadratischen Puzzle-Teile sollen ausgeschnitten und wieder so in einem 3x3-Muster angeordnet werden, dass alle inneren Mechatronik-Logos korrekt zueinander passen, d.h. die beiden Hälften der Beschriftung und die Farben des Randes müssen jeweils zueinander passen. Die einzelnen Puzzle-Teile dürfen dabei in Schritten von 90 Grad gedreht werden.

Es gibt nur begrenzt viele Kombinationsmöglichkeiten, die Karten auf die geforderte Art hinzulegen. Insgesamt sind es  $9! \cdot 4^9 = 95126814720$  Möglichkeiten, und jeder von ihnen ist leicht anzusehen, ob sie eine Lösung darstellt. Es ist also Ihnen überlassen, ob Sie hierbei einfach entspannt vor sich hin probieren, oder Sie dieses als abstraktes Problem der Informatik auffassen und sich ein systematisches Lösungsverfahren überlegen, das möglichst schnell und intelligent eine Lösung des Puzzles finden soll. Falls Sie einen Computer besitzen und Programmierkenntnisse haben, dann können Sie einen solchen Algorithmus auch in ein Programm umsetzen, das Ihnen eine Lösung berechnet. Viel Spaß! ■

