

## Informatik und allgemeine Bildung

Heutzutage verbreiten sich Computer in den Haushalten so wie einst Telefone oder Fernseher. PC, Internet, E-Mail und WWW sind nur einige Stichworte, die im Zusammenhang mit der Informatik in der Öffentlichkeit häufig benutzt werden. Aber ein erfolgreicher Umgang mit Personal Computern oder «Surfen» im Internet hat noch aus niemandem einen Informatiker gemacht, genauso wenig, wie das Autofahren alleine Maschinenbauer oder das Nutzen elektrischer Geräte Physiker hervorbringt. Die Öffentlichkeit und mit ihr ein Grossteil der Führungskräfte in unserer Gesellschaft haben eine ziemlich falsche Vorstellung von der Informatik als wissenschaftliche Disziplin. Die Konsequenz ist, dass man die Rolle der Informatik in der allgemeinen Bildung oft unterschätzt und in mehreren Bildungskonzepten auf oberflächliche Computer-Nutzung reduziert. Zielsetzung dieses Artikels ist es, mehr Verständnis für die Natur der Informatik zu vermitteln und auf die möglichen prinzipiellen Beiträge zum Allgemeinwissen, insbesondere auf revolutionäre Änderungen in der Denkweise, aufmerksam zu machen. Die in der Informatik erworbene Kompetenz, mit Maschinen zu kommunizieren, soll auch in den Vordergrund gehoben werden.

Versuchen wir zunächst, die Frage «**Was ist Informatik?**» zu beantworten. Eine genaue Spezifikation einer wissenschaftlichen Disziplin zu liefern, ist eine schwierige Aufgabe, die man selten vollständig bewältigen kann. Üblicherweise wird versucht, Informatik mit der folgenden allgemeinen Aussage zu beschreiben:

*«Informatik ist die Wissenschaft von der algorithmischen Darstellung, Erkennung, Verarbeitung, Speicherung und Übertragung von Information.»*

Obwohl diese meist akzeptierte Definition der Informatik die Information und den Algorithmus als Hauptobjekte und den Umgang mit diesen als Ziel der Untersuchung in der Informatik darstellt, sagt sie nicht viel über die Natur der Informatik und über die in ihr benutzten Methoden aus. Eine viel wichtigere Frage für die Klärung der Substanz der Informatik ist die folgende:

*«Welchen Wissenschaften kann man die Informatik zuordnen? Ist sie Meta-Wissenschaft (wie Philosophie und Mathematik), Geisteswissenschaft, Naturwissenschaft oder Ingenieurwissenschaft?»*

Die Antwort auf diese Frage klärt nicht nur das Objekt der Untersuchung, sondern sie legt auch die Methodik und die Beiträge der Informatik fest. Die Antwort lautet, dass die Informatik keiner dieser Wissenschaftsgruppen vollständig zugeordnet werden kann. Die Informatik besitzt sowohl die Aspekte einer Meta-Wissenschaft, einer Naturwissenschaft als auch die einer Ingenieurwissenschaft. Wir geben hier eine kurze Begründung für diese Behauptung.

Wie die Philosophie und die Mathematik studiert die Informatik allgemeine Kategorien wie

*Determinismus, Nichtdeterminismus, Zufall, Information, Wahrheit, Unwahrheit, Komplexität, Sprache, Beweis, Wissen, Kommunikation, Approximation, Algorithmus, Simulation usw.*

und trägt zu ihrem Verständnis bei. Mehreren dieser Kategorien hat die Informatik einen neuen Inhalt und eine neue Bedeutung gegeben.

Eine Naturwissenschaft studiert (im Unterschied zur Philosophie und Mathematik) konkrete physikalische Objekte und Prozesse, bestimmt die Grenze zwischen Möglichem und Unmöglichem und erforscht die quantitativen Gesetze der Naturprozesse. Die Naturwissenschaften modellieren also die Realität, analysieren die gewonnenen Modelle und überprüfen ihre Zuverlässigkeit in Experimenten. Alle diese Aspekte einer Naturwissenschaft finden sich auch in der Informatik wieder. Die Objekte sind Informationen und Algorithmen (Programme, Rechner), und die Prozesse sind die physikalisch existierenden Prozesse der Informationsverarbeitung. Versuchen wir, dies anhand der Entwicklung der Informatik zu dokumentieren. Die historisch erste wichtige Forschungsfrage der Informatik war die folgende Frage von philosophischer Bedeutung:

*Existieren wohl definierte Aufgaben, die man automatisch (d.h. durch einen Rechner, unabhängig von der Leistungsfähigkeit heutiger oder zukünftiger Rechner) nicht lösen kann?*

Die Bemühungen, diese Frage zu beantworten, führten zur Gründung der Informatik als selbständige Wissenschaft. Die Antwort auf diese Frage ist positiv, und wir kennen heute viele praktisch relevante Aufgaben, die man gerne algorithmisch (automatisch) lösen würde, die aber algorithmisch nicht lösbar sind. Das liegt aber nicht daran, dass bisher niemand einen Algorithmus (ein Programm) zur Lösung dieser Aufgaben entwickelt hat, sondern daran, dass die Nichtexistenz solcher Programme mathematisch bewiesen wurde.

Nachdem man Methoden entwickelt hat, um Aufgaben diesbezüglich zu klassifizieren, ob für diese ein Programm als algorithmische Lösung existiert oder nicht, stellt man sich die naturwissenschaftliche Frage: «**Wie schwer sind konkrete algorithmische Aufgaben?**»

Die Schwierigkeit einer Aufgabe misst man aber nicht darin, wie schwer es ist, ein Programm für die Aufgabe zu entwickeln, oder wie umfangreich solch ein Programm ist. Die Schwierigkeit misst man in der Menge der Arbeit, die ein Rechner leisten muss, um die Aufgabe für konkrete Eingaben zu lösen. Man hat festgestellt, dass es beliebig schwere Aufgaben gibt, sogar solche, für deren Lösung man mehr Energie braucht, als im ganzen bekannten Universum zur Verfügung steht. Es existieren also Aufgaben, für deren Lösung man zwar Programme schreiben kann, was aber nicht hilft, denn ein Lauf eines solchen Programms benötigt mehr Zeit, als etwa seit dem Urknall bis heute vergangen ist. Die bloße Existenz eines Programms für eine untersuchte Aufgabe bedeutet also nicht, dass diese Aufgabe **praktisch** algorithmisch lösbar ist.

Die Bemühungen, die Aufgaben in **praktisch lösbare** und **praktisch unlösbare** zu unterteilen, führten zu einigen der faszinierendsten mathematisch-naturwissenschaftlichen Erkenntnisse, die in der Informatik erbracht worden sind.

Als ein Beispiel solcher Resultate können wir zufallsgesteuerte Algorithmen betrachten<sup>1</sup>. Programme (Algorithmen), wie wir sie benutzen, sind deterministisch. Die Bedeutung des Wortes «deterministisch» besteht darin, dass das Programm und die Eingabe vollständig die Arbeit des Rechners bestimmen. Zu jedem Zeitpunkt ist in Abhängigkeit von den aktuellen Daten eindeutig bestimmt, was die nächste Aktion des Programms sein wird. Zufallsgesteuerte Programme dürfen mehrere Möglichkeiten für die Fortsetzung ihrer Arbeit aufweisen; welche Möglichkeit gewählt wird, wird zufällig entschieden. Das ist so, als würde ein zufallsgesteuertes Programm von Zeit zu Zeit eine Münze und wählte abhängig davon, ob Kopf oder Zahl gefallen ist, eine entsprechende Strategie für die weitere Verfahrensweise, etwa zur Suche nach einem richtigen Resultat. Ein zufallsgesteuertes Programm erlaubt daher mehrere unterschiedliche Berechnungen für ein und dieselbe Eingabe. Im Unterschied zu deterministischen Programmen, die immer eine zuverlässige Berechnung des richtigen Resultats garantieren, dürfen einige Berechnungen zufallsgesteuerter Programme auch zu falschen Resultaten führen. Das Ziel besteht darin, die Wahrscheinlichkeit der Durchführung einer Berechnung mit falschem Resultat so gering wie möglich zu halten, was unter bestimmten Umständen bedeuten kann, dass man versucht, den proportionalen Anteil der Berechnungen mit falschem Resultat zu minimieren.

Auf den ersten Blick sieht ein zufallsgesteuertes Programm im Vergleich zu deterministischen Programmen wie etwas Unzuverlässiges aus, und man könnte sich fragen, wozu das gut sein soll. Es existieren aber Aufgaben von grosser praktischer Bedeutung, bei denen der schnellste deterministische Algorithmus auf dem schnellsten denkbaren Rechner mehr Zeit zur Berechnung der Lösung bräuchte, als das Universum alt ist. Die Aufgabe scheint also, praktisch unlösbar zu sein. Und nun geschieht ein «Wunder»: Ein zufallsgesteuerter Algorithmus, der die Aufgabe in ein paar Minuten auf einem gewöhnlichen Personal Computer mit einer Fehlerwahrscheinlichkeit von einem Billionstel löst. Kann man ein solches Programm für unzuverlässig halten? Der Lauf eines deterministischen Programms, das für eine Aufgabe während eines Tags rechnet, ist unzuverlässiger als unser zufallsgesteuertes Programm, weil die Wahrscheinlichkeit des Auftretens eines Hardware-Fehlers während einer 24-stündigen Arbeit viel höher ist als die Wahrscheinlichkeit einer fehlerhaften Ausgabe des schnellen zufallsgesteuerten Programms.

Warum so etwas überhaupt möglich ist, ist ohne Informatikvorkenntnisse schwer zu erklären. Die Suche nach den wahren Gründen für die Stärke der Zufallssteuerung ist aber eine faszinierende mathematisch-naturwissenschaftliche Forschungsaufgabe. Wichtig ist, zu bemerken, dass auch hier die Natur unser bester Lehrmeister sein kann, denn in ihr geschieht mehr zufallsgesteuert, als man glaubt. Informatiker können viele Beispiele von Systemen angeben, bei denen die gewünschten Eigenschaften und Verhaltensweisen im Wesentlichen durch das Konzept der Zufallssteuerung erreicht werden. In solchen Beispielen muss jedes deterministische, «einhundert Prozent zuverlässige» System mit dem erwünschten Verhalten aus Milliarden von Teilsystemen bestehen, die alle miteinander kooperieren müssen. Ein solch komplexes System, bei dem viele Teilsysteme immer korrekt arbeiten, kann praktisch nicht realisiert werden, und falls ein Fehler auftritt, ist es eine fast unlösbare Aufgabe, ihn zu suchen. Man braucht gar nicht darüber nachzudenken, wie hoch die Entwicklungs- und Herstellungskosten eines solchen Systems wären. Andererseits kann man ein solches System zu geringen Kosten durch ein kleines zufallsgesteuertes System mit dem gewünsch-

---

<sup>1</sup> Es ist eine Vorliebe des Autors, gerade dieses Beispiel zu verwenden. Die Informatik besitzt eine Menge eindrucksvoller Konzepte, die einen erstaunen lassen.

ten Verhalten ersetzen, bei dem alle Funktionen jederzeit überprüfbar sind und die Wahrscheinlichkeit eines fehlerhaften Verhaltens so gering ist, dass man sich in der Anwendung keine Sorgen darüber machen muss.

Trotz der naturwissenschaftlichen Aspekte der Informatik, die wir gerade illustriert haben, bleibt diese für die meisten Informatiker eine typische anwendungs- und problemorientierte Ingenieurwissenschaft. Die Informatik umfasst nicht nur die technischen Aspekte des Ingenieurwesens, wie

*Organisation des Entwicklungsprozesses (Phasen, Meilensteine, Dokumentation), Formulierung strategischer Ziele und einzuhaltender Grenzen, Modellierung, Beschreibung, Spezifikation, Qualitätssicherung, Testen, Einbettung in existierende Systeme, Wiederverwendung und Werkzeugunterstützung,*

sondern auch die Management-Aspekte wie zum Beispiel

*Team-Organisation und -Leitung, Kostenvoranschlag und Kostenaufschlüsselung, Planung, Produktivität, Qualitäts-Management, Abschätzung von Zeitrahmen und Fristen, Zeit zur Markteinführung, Vertragsabschluss und Marketing.*

Eine Informatikerin oder ein Informatiker muss auch ein wahrer Pragmatiker sein. Bei der Konstruktion sehr komplexer Soft- oder Hardware-Systeme muss man Entscheidungen oft gefühlsmässig und anhand eigener Erfahrung treffen, weil man keine Aussicht hat, die komplexe Realität vollständig zu analysieren und zu modellieren.

Ein anschauliches Beispiel ist das Konstruieren grosser Software-Systeme, die aus mehr als einer Million Recheninstruktionen bestehen. Es ist angesichts derart langer und komplexer Programme praktisch unmöglich, ein fehlerfreies Produkt herzustellen, insbesondere wenn an seiner Herstellung eine grosse Anzahl von Software-Ingenieuren beteiligt ist. Also weiss man im Vorhinein, dass ein fehlerhaftes Verhalten des Endproduktes nicht zu vermeiden ist, und es geht nur darum, ob die Anzahl dieser Fehler sich in Hunderten, Tausenden oder sogar Zehntausenden messen lassen muss. Und da kommen die meisten der oben erwähnten Konzepte ins Spiel. Gute Modellierung, Strukturierung und Spezifikation in der Herstellungsphase und geschicktes Verifizieren und Testen während des ganzen Projekts sind entscheidend für die endgültige Zuverlässigkeit des Software-Produkts.

Wenn man sich das, was wir bisher von der Informatik geschildert haben, durch den Kopf gehen lässt, könnte man den Eindruck gewinnen, dass das Studium der Informatik zu schwer sei. Gute Mathematikkenntnisse sind erforderlich, und die Denkweise von Naturwissenschaftlern und Ingenieuren ist gleichermassen erwünscht. Das mag stimmen, aber in dieser Interdisziplinarität liegt auch der grösste Vorteil dieser Ausbildung. Die grösste Krankheit heutiger Wissenschaften ist eine zu starke Spezialisierung, die dazu geführt hat, dass sich viele Wissenschaften zu unabhängig voneinander entwickelt haben. Die Wissenschaften entwickelten eigene Sprachen, die oft sogar für benachbarte Wissenschaften nicht mehr verständlich sind. Es geht soweit, dass die standardisierte Art der Argumentation in einer Wissenschaft in einer anderen Wissenschaft als eine oberflächliche und unzulässige Begründung eingestuft wird. Das macht die propagierte interdisziplinäre Forschung ziemlich schwierig. Die Informatik aber ist in ihrem Kern interdisziplinär. Sie orientiert

sich an der Suche nach Problemlösungen in allen Bereichen des wissenschaftlichen und alltäglichen Lebens, bei denen man Rechner anwendet oder anwenden könnte. Dabei bedient sie sich eines breiten Spektrums von Verfahren, das von präzisen formalen Methoden der Mathematik bis hin zum erfahrungsgetriebenen «Know-how» der Ingenieurdisziplinen variiert. Die Möglichkeit, gleichzeitig unterschiedliche Wissenschaftssprachen und Arten des Denkens zusammenhängend in einer Disziplin zu erlernen, ist das wichtigste, was den Informatikabsolventen in ihrer Ausbildung vermittelt wird.

Dies führt uns zur Kernfrage dieses Abschnitts, zur Frage, wie man die Informatik in die allgemeine Bildung einfließen lassen kann. Dabei ist wichtig, zu bemerken, dass es hier nicht darum gehen soll, wie man erfolgreich Rechner zur Unterstützung der Ausbildung in anderen Fächern nutzt, sondern darum, welche Teile der Kerninformatik man wie und wo in der allgemeinen Bildung anbieten kann.

Die Informatik mit ihren Erkenntnissen mathematischer und naturwissenschaftlicher Natur hat heute für die allgemeine Bildung eine genauso grosse Bedeutung wie jedes andere Fach, das fester Bestandteil der Ausbildung ist. Am besten sind die potenziellen Bildungsbeiträge der Informatik mit jenen der Mathematik zu vergleichen, nicht nur, weil die angewandten Methoden mathematischer Natur sind und viele Erkenntnisse in der Schnittmenge von Mathematik und Informatik liegen, sondern insbesondere auch, weil die Kenntnisse der Kerninformatik ähnlich allgemeine Bedeutung für Lösungen von Problemen haben und genauso wichtig für die Prägung von Denkweisen sind wie mathematische Kenntnisse. Beispiele dafür sind nicht nur die oben erwähnten zufallsgesteuerten Programme und Systeme, sondern auch viele abstrakte Anwendungen in der Kommunikation, insbesondere im geheimen Informationsaustausch (Kryptographie). Das Schöne an der Präsentation dieser Gebiete ist, dass man die Ziele als aufregende Probleme erfährt und auf der Lösungsebene viele unerwartete Wendungen und Überraschungen erleben kann. Dies ist gerade das, was der heutigen Mathematikausbildung in Mittelschulen fehlt. Informatikunterricht kann in mehreren Dimensionen eine Bereicherung und Unterstützung des Mathematikunterrichts sein und soll in keinem Fall als Konkurrenz für die Mathematik betrachtet werden. Man kann sich zum Beispiel der Entwicklung von algorithmischen Strategien zur Lösung mathematischer Probleme widmen. Solche Zielsetzungen sind leicht zu motivieren. Darüber hinaus kann man die entwickelten Methoden und Strategien auf dem Rechner und oft sogar im täglichen Leben umsetzen und testen. Solche experimentierfreudigen und puzzle-artigen Elemente brächten wieder mehr Leben in den Mathematikunterricht. Die anschauliche Geometrie kann sehr früh erfolgreich vermittelt werden, wenn man im Programmierunterricht im Primarschulalter mit Programmen zur Zeichnung von geometrischen Bildern anfängt.

Es verbleibt natürlich die Frage, ob und wie man die praktischen, ingenieurwissenschaftlichen Teile der Informatik in die allgemeine Ausbildung einbringen kann. Diese sind wesentlich bedeutsamer als das Erlernen des Umgangs mit dem Rechner. Zum Beispiel bedeutet Programmieren, ein gewünschtes Verhalten oder Tätigkeiten auf der Ebene einfacher Aktionen so eindeutig wie möglich und in der Form eines Rezepts zu beschreiben, dass nicht nur jeder nicht eingeweihte Mensch, sondern sogar die Maschine, bei jeder Intelligenz, die gewünschte Tätigkeit nach dem vorgelegten Rezept erfolgreich ausüben kann. Diese Fertigkeit wird im täglichen Leben mit der Technik mehr und mehr gefragt, denn sie entspricht in gewissem Sinne der Fähigkeit, mit den Maschinen zu kommunizieren. Es ist sicher nicht zu unterschätzen, wenn Kinder und Jugendliche eine grosse

Freude am Programmieren entwickeln. Das rührt daher, dass sie etwas entwerfen dürfen, das sie danach testen und durch Fehlersuche korrigieren und verbessern können. Etwas, das vielleicht für viele nicht glaubhaft klingen mag, ist, dass es Grundschulkindern mit Begeisterung gelingt, die Grundlagen der Programmierung zu erlernen und selbständig anzuwenden. Einige Zukunftsvisionen schreiben dem Algorithmenentwurf und der Programmierung eine dem Lesen und Schreiben vergleichbare Bedeutung zu. Ob das wirklich so wörtlich zu nehmen ist, wird die Zeit zeigen. Aber in der Epoche der Rechner und breiter Kommunikationsmöglichkeiten ist gerade die Informatik die Disziplin, die das tiefere Verständnis für all diese Prozesse der Informationsverarbeitung und -übertragung vermitteln kann.

Wir haben mehrere unterschiedliche Gründe für eine stärkere Einbeziehung der Informatik in die allgemeine Bildung genannt. Der wichtigste liegt wahrscheinlich in der Interdisziplinarität, die das innere Merkmal der Informatik ist. Max Planck hat einmal gesagt, dass es nur eine Wissenschaft gibt und dass ihre Verteilung in unterschiedliche Wissenschaftsdisziplinen der beschränkten menschlichen Kapazität, aber nicht ihrem inneren Bedarf entspricht. Die Informatik ist ein «lebendiger» Beweis der Gültigkeit dieser Behauptung. Informatik könnte gerade den fehlenden Integrationsfaktor zwischen verschiedenen Wissenschaftsgebieten liefern und so zur Modernisierung des ganzen Bildungssystems beitragen.

Mit Integration der Informatik geht es nicht nur darum, die schon vor vielen Jahren entstandene Lücke in der allgemeinen Ausbildung zu schliessen, sondern einen wesentlichen Schritt nach vorne in der Entwicklung unseres Schulsystems zu machen.

Juraj Hromkovic

### **Buchhinweise**

- Freiermuth, Karin; Hromkovic, Juraj; Keller Lucia; Steffen, Björn: Einführung in die Kryptologie. Lehrbuch für Unterricht und Selbststudium. Vieweg + Teubner Verlag, Wiesbaden, 1. Auflage 2010, 407 Seiten,
- Hromkovic, Juraj: Algorithmic Adventures. From Knowledge to Magic. Springer-Verlag, Heidelberg 2009, 380 Seiten,
- Hromkovic, Juraj: Einführung in die Programmierung mit Logo. Lehrbuch für Unterricht und Selbststudium. Vieweg + Teubner Verlag, Wiesbaden 2010, 272 Seiten,
- Hromkovic, Juraj: Lehrbuch Informatik. Vorkurs Programmieren, Geschichte und Begriffsbildung, Automatenentwurf. Vieweg + Teubner Verlag, Wiesbaden 2008, 512 Seiten,
- Hromkovic, Juraj: Sieben Wunder der Informatik. Eine Reise an die Grenze des Machbaren mit Aufgaben und Lösungen. Vieweg + Teubner Verlag, Wiesbaden 2009, 360 Seiten.

Stand: 1.5.2010