

Theoretische Informatik – ein Kurzprofil

Wolfgang Thomas

Die Anfänge

Vieles von dem, was wir heute Theoretische Informatik nennen, reicht zurück in die Zeit vor der Etablierung des Studienfachs Informatik um 1970. Damals waren einige theoretische Grundlagen bereits Jahrzehnte alt: Die Theorie der Berechenbarkeit (und damit die Bestimmung der Grenzen algorithmischer Methoden) war schon in den 1930er Jahren durch Forscher wie Kurt Gödel, Alonzo Church und Alan Turing entwickelt worden. Das Modell des endlichen Automaten hatte sich in den 1950er Jahren herausgebildet (durch Stephen Kleene, Michael Rabin, Dana Scott und andere), ebenso die Grundlegung der Theorie der Grammatiken (Noam Chomsky). Ein weiterer Schub erfolgte dann in den 1970er Jahren, als sich daneben die Komplexitätstheorie u.a. mit der Konzeption der Problemklassen P und NP und die Fundamente der Algorithmik, insbesondere mit dem Feld der Graphalgorithmen, entwickelten. Parallel dazu bewährte sich die Theorie der Automaten und formalen Sprachen als Grundlage für den Übersetzerbau.

Der zugehörige Corpus an Begriffen, Resultaten und Methoden bildet den Inhalt zahlreicher Lehrbücher zur Theoretischen Informatik. Wegweisend war unter anderem das Buch von J.E. Hopcroft und J.D. Ullman (*Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley 1980). Der Stoffkanon „Automaten und formale Sprachen, Berechenbarkeit, Komplexität“ hat lange Zeit den gemeinsamen Nenner der theoretischen Ausbildung der Informatiker gebildet und damit auch sehr zum wissenschaftlichen Selbstverständnis der Informatik beigetragen.

Doch sind inzwischen so viele neue Themen, Gebiete, Methoden und vor allem Ergebnisse hinzugekommen, dass die klassischen Grundlagen nur noch als das „kleine Einmaleins“ der Theoretischen Informatik zählen können. Dieses vermittelt heute keinen Eindruck mehr von der Vielfalt, welche die Theoretische Informatik inzwischen bietet.

Inhalte

Es soll hier nicht versucht werden, eine vollständige Übersicht über die aktuellen Inhalte der Theoretischen Informatik zu geben. Dennoch können einige Teildisziplinen genannt werden. Zunächst betrachten wir die Zeit vor etwa 20 Jahren, die heute auch schon wieder „historisch“ ist. Bereits damals hatte sich der Bereich der Theoretischen Informatik schon wesentlich erweitert. Eine nützliche Quelle, die den Stand zu jener Zeit illustriert, ist das *Handbook of Theoretical Computer Science* (herausgegeben von J. v. Leeuwen, Elsevier 1990), das in etwa 40 Einzelartikeln wichtige Forschungsgebiete beschreibt. Das Handbuch ist in zwei Bände gegliedert. Für diejenigen Leserinnen und Leser, die durch Aufzählungen nicht abgeschreckt werden, geben wir eine Titelübersicht in deutscher Übersetzung:

Band A (Algorithmen and Komplexität): Maschinenmodelle und Simulationen, Ein Katalog der Komplexitätsklassen, Maschinenunabhängige Komplexitätstheorie, Kolmogorov-Komplexität und ihre Anwendungen, Algorithmen für Mustererkennung in Wörtern, Datenstrukturen, Algorithmische Geometrie, Algorithmische Wegeplanung in der Robotik, Average-Case-Analyse von Algorithmen und Datenstrukturen, Graphalgorithmen,

Algebraische Komplexitätstheorie, Algorithmen in der Zahlentheorie, Kryptographie, Komplexität endlicher Funktionen, Kommunikationsnetzwerke, VLSI-Theorie, Parallele Algorithmen für Maschinen mit gemeinsamem Speicher, Parallele Architekturen.

Band B (Formale Modelle und Semantik): Endliche Automaten, Kontextfreie Sprachen, Formale Sprachen und Potenzreihen, Automaten auf unendlichen Objekten, Graph-Ersetzung: ein algebraischer und logischer Ansatz, Ersetzungssysteme, Funktionale Programmierung und Lambda-Kalkül, Typsysteme für Programmiersprachen, Rekursive Programmschemata, Logik-Programmierung, Denotationelle Semantik, Semantische Bereiche, Algebraische Spezifikation, Programmlogiken, Methoden und Logiken für Programmverifikation, Temporale und modale Logik, Elemente relationaler Datenbanktheorie, Verteiltes Rechnen: Modelle und Methoden, Operationale und algebraische Semantik nebenläufiger Prozesse.

Die in diesem Handbuch vorgenommene Einteilung der Theoretischen Informatik in die beiden Säulen „A. Algorithmik“ und „B. Formale Modelle“ hat prägend gewirkt. Sie zeigt sich zum Beispiel in der Struktur der führenden europäischen Tagung ICALP zur Theoretischen Informatik, die in die „Tracks“ A und B gegliedert ist (seit einigen Jahren ergänzt durch einen „Track C“ für wechselnde Themen von aktuellem Interesse). Auch wird die Theoretische Informatik in manchen Informatik-Fachbereichen durch Lehrstühle mit den Orientierungen A und B vertreten.

In den vergangenen zwei Jahrzehnten sind zahlreiche neue Forschungsfelder hinzugekommen. Hierunter fallen die Gebiete Approximationsalgorithmen, Online-Algorithmen, randomisierte Algorithmen, Algorithmische Spieltheorie, Model-Checking, Quantum Computing und viele mehr. Über einige dieser Gebiete finden sich Beschreibungen im vorliegenden Themenheft.

Zwecke

Die allgemeine Aufgabe der Theoretischen Informatik mag man wie folgt umreißen:

Sie soll erstens Begriffe, Methoden und Ergebnisse bereitstellen, welche die prinzipiellen Möglichkeiten der Informationsdarstellung und -verarbeitung klären. So bemühen sich Berechenbarkeitstheorie und Komplexitätstheorie um die gleichsam naturgesetzlichen Bedingungen, unter denen die Informatik arbeitet. Solche Analysen zu den prinzipiellen Möglichkeiten sind notwendig, um Modelle oder Beschreibungsformalismen tiefer zu verstehen (zum Beispiel um Möglichkeiten der Anwendung zu beurteilen). So werden Algorithmen, Programmiersprachen, Berechnungsmodelle, Logiken usw. daraufhin untersucht, was sich damit jeweils berechnen, ausdrücken oder auch nur rechnerisch analysieren lässt.

Ein anderer Zweck ist die Bereicherung des Spektrums an allgemeinen Methoden zur Lösung informatischer Probleme. Viele Kapitel der Algorithmik dienen dem ganz praktischen Zweck, effiziente und umfassend anwendbare Hilfsmittel zur Problemlösung zu entwickeln. Ähnlich wird überall dort, wo Logik-Formalismen zur Spezifikation oder Verifikation eingesetzt werden, nach einer Balance zwischen Ausdruckskraft einerseits und algorithmischer Handhabbarkeit bzw. praktischer Anwendbarkeit andererseits gesucht.

Beide Zwecke greifen ineinander und führen in ihrer Verbindung zur zentralen Aufgabe der Theoriebildung, häufig beginnend mit der Formung spezifischer Begriffe und Modelle. Über

dieses „Hauptgeschäft“ der Theoretischen Informatik ließe sich sehr ausführlich diskutieren, insbesondere zur Frage, wann eine Theorie nun „gut“ ist. Dies soll hier nicht weiter ausgeführt werden, doch ist klar, dass der Wert einer Theorie nicht nur nach der unmittelbaren Anwendbarkeit beurteilt werden kann, sondern dass auch die gedankliche Eleganz, die Schwierigkeit der gelösten Probleme und die Ausstrahlung auf benachbarte Felder wesentliche Kriterien sind.

Das Vorgehen ist häufig wie folgt: Ausgangspunkt ist beispielsweise ein Berechnungsmodell (ein Automatenmodell, ein Algorithmientyp, eine Programmiersprache). Es werden Begriffe und Methoden konzipiert, die für die Analyse des Berechnungsmodells geeignet sind und die dann zu allgemeinen Resultaten über dieses Modell führen. Zur Illustration sei etwa das Beispiel der Approximationsalgorithmen betrachtet: Der Begriff der Approximationsgüte liefert ein nützliches Kriterium zur Bewertung der Qualität solcher Algorithmen. Es gibt dann Methoden, Approximationsalgorithmen bestimmter Güte und Laufzeit zu entwickeln. Schließlich lässt sich für Klassen von Berechnungsproblemen auch eingrenzen, welche Approximationsgüte überhaupt (durch einen effizienten Algorithmus) erreichbar ist.

Der allgemeine Wunsch nach „Effizienz“ darf aber nicht darüber hinwegtäuschen, dass auch der Nachweis prinzipieller Grenzen, etwa durch untere Schranken für die Komplexität von Berechnungsproblemen, eine große praktische Bedeutung haben kann. Dies gilt insbesondere für das Feld der Kryptographie; hier waren tiefgehende Analysen der Komplexitätstheorie Wegbereiter für die Konzeption äußerst erfolgreicher Verschlüsselungsmethoden. Der in diesen Forschungen etablierte Begriff der „zero knowledge proofs“ (mit denen man die Kenntnis von Informationen begründen kann, ohne sie zu verraten) hat darüber hinaus auch erkenntnistheoretische Bedeutung.

Und schließlich besteht – wie später noch genauer erläutert wird - ein wesentliches Ziel der Theoretischen Informatik darin, die gemeinsamen wissenschaftlichen Fundamente der Informatik-Teildisziplinen weiter zu entwickeln. Dazu gehört besonders die permanente Arbeit an den begrifflichen Grundlagen, so wie sie in der Informatik-Lehre vermittelt werden sollen.

Fließende Grenzen

Die Dynamik der Theoretischen Informatik wird an vielen Stellen durch die Aufweichung oder gar Auflösung von Abgrenzungen gefördert, sowohl im „Inneren“ als auch im Verhältnis zu anderen Disziplinen. Wir erwähnen nur einige Beispiele für diesen Trend der Integration.

So bequem die Einteilung der Theoretischen Informatik in die oben genannten Säulen A und B auch sein mag – die aktuelle Entwicklung ist auch gegenläufig, indem scheinbar weit entfernte Bereiche zusammengeführt werden. Die „deskriptive Komplexitätstheorie“ ist ein Beispiel: Sie zeigt, dass sich Komplexitätsklassen wie NP oder PSPACE durch Logiken charakterisieren lassen. Komplexitätsklassen sind damit sowohl in der algorithmischen als auch in der logischen Sicht behandelbar. Auch sind zentrale Konzepte der Datenbanktheorie, die man vielleicht eher den „formalen Modellen“ zuordnen möchte, sehr eng mit der algorithmischen Graphentheorie verknüpft.

Eine andere in Bewegung geratene Grenze betrifft das Verständnis dessen, was ein „schweres Berechnungsproblem“ ist. Eine oberflächliche, in vielen Lehrbüchern gepflegte Sicht besagt, dass die NP-vollständigen Probleme dazugehören. Doch hat die Forschung der letzten zwei Jahrzehnte den Schrecken vor der NP-Vollständigkeit ein gutes Stück abgebaut

und gezeigt, dass man hier viel feiner unterscheiden muss - etwa nach Approximierbarkeit von Lösungen, nach verschiedenen Graden exponentieller Laufzeit, nach Unterscheidungen mit Techniken der probabilistischen Analyse. Auch hat die Entwicklung der SAT-Solver, der ILP-Solver (für ganzzahlige lineare Programmierung) und der Theorembeweiser die Einschätzung dessen, was sich praktisch berechnen lässt, verschoben; vgl. hierzu die Beiträge von U. Schöning sowie von F. Baader, B. Beckert und T. Nipkow im vorliegenden Heft.

Forschungen in diesen Bereichen sind ohne ausgedehnte praktische Untersuchungen nicht denkbar. In der Theoretischen Informatik wird also nicht nur mathematisch-deduktiv gearbeitet, sondern zunehmend auch gestützt auf Experimente. Mit der Disziplin des Algorithm Engineering hat sich inzwischen eine eigene Methodik zur Algorithmenentwicklung herausgebildet, in welcher Theorie und Experiment zusammenspielen (vgl. den Artikel von P. Sanders im vorliegenden Heft). Es gibt also keine strenge Grenze mehr, welche das Vorgehen der Theoretischen Informatik vom Ansatz der experimentellen Wissenschaften grundsätzlich trennen würde.

Zu den Nachbarwissenschaften, insbesondere zur Mathematik, sind die Grenzen ebenfalls fließend. Dies ist vor allem sichtbar in der rasanten Entwicklung der diskreten Mathematik und der Theorie der Optimierung, die sich in weiten Bereichen mit den algorithmischen Zweigen der Theoretischen Informatik überschneiden. Die Computeralgebra ist eine weitere Disziplin, in der sich die Anteile aus Mathematik und Informatik kaum trennen lassen. Wie sehr die Mathematik inzwischen die Fragestellungen der Theoretischen Informatik aufgreift, sieht man an der Geschichte des P-NP-Problems: Es entstand in der Theoretischen Informatik, wurde über Jahrzehnte hin mehr und mehr als Problem auch der Mathematik wahrgenommen, und es erschien im Jahre 2000 schließlich in der Liste der sieben prominentesten mathematischen Probleme, der „Millenium Prize Problems“ des Clay Institute for Mathematics, Seite an Seite mit der Riemannschen Hypothese.

Schließlich sollten auch die enger werdenden Beziehungen zur Physik und zur Biologie genannt werden. In der Tat lassen sich viele Fragen über große Netzwerke nicht ohne einen Blick auf die Forschungen etwa der statistischen Physik oder der Neurowissenschaften behandeln. Auch nicht-klassische Berechnungsparadigmen wie das „Quantum Computing“ oder „DNA-Computing“ erfordern die enge Kooperation mit Physikern bzw. Biologen.

Herausforderungen

Natürlich gilt es, herausragende offene Probleme wie das genannte P-NP-Problem zu lösen, so schwierig sie auch sein mögen. Aber auch jede einzelne Teildisziplin der Theoretischen Informatik bietet eine Fülle offener Fragen. Wer sich über wichtige Fortschritte ein Bild machen will, kann sich etwa an den Verleihungen des Gödel-Preises orientieren. Dieser Preis für herausragende Resultate in der Theoretischen Informatik wird jährlich gemeinsam von den Fachgesellschaften EATCS (European Association for Theoretical Computer Science) und ACM SIGACT (Association of Computing Machinery, Special Interest Group on Algorithms and Computation Theory) vergeben.

Eine allgemeinere Art von Herausforderung besteht darin, für Schlüsselfragen der Informatik angemessene Methoden zu entwickeln. Zu den fundamentalen methodischen Problemen der Informatik seien nur drei Fragenkreise als Beispiele herausgegriffen: Wie sieht eine „realistische Theorie der Berechnungskomplexität“ aus, die aus praktischer Sicht die grobe Klassifizierung nach asymptotischer Worst-Case-Abschätzung übertrifft? Wie lassen sich die

die Modellierungen diskreter und kontinuierlicher Systeme integrieren, auch unter Verwendung von Begriffen der Stochastik? Wie ist eine handhabbare und aus praktischer Sicht adäquate Theorie strukturierter (z.B. hierarchischer, verteilter) Systeme zu gestalten? Für alle drei Fragenkomplexe gibt es bemerkenswerte Erfolge (in unseren Beispielen etwa mit der „Smoothed Analysis“ bzw. der entstehenden Theorie der Realzeit-Systeme und hybrider Systeme bzw. Theorien der „Concurrency“), doch erscheint der Weg noch weit bis zu eleganten und wirklich umfassenden Theorien.

Ganz andersartige Herausforderungen stellen sich jenseits offener Fachfragen auf der „Metaebene“, im Zusammenspiel zwischen Theoretischer Informatik und dem „Rest der Informatik“.

Für die Informatik sind erfolgreiche Theoriebildungen sicherlich wichtig, damit die eher angewandten Forschungszweige befruchtet werden. Und wie alle Wissenschaftsdisziplinen steht auch die Forschung in der Theoretischen Informatik unter dem Druck, durch zahlreiche Forschungsprojekte möglichst viele Drittmittel einzuwerben. Anwendbarkeit ist hier das dominierende Ziel. Dies fördert zweifellos die Forschungsdynamik, oft auch den Kontakt etwa zur Praktischen Informatik – beides sehr positive Effekte.

Doch drohen hier durchaus Gefahren: Es kann daneben die Erfüllung der zentralen Aufgabe der Theorie leiden, die begrifflichen und methodischen Grundlagen der Informatik zu erneuern. Dieses mühsame Geschäft der „Aufarbeitung“ ist entscheidend für die Profilierung der Informatik als Wissenschaft, gerade angesichts rascher Fortschritte in ihren vielen Teilgebieten. Die immer wieder neu betriebene Ausgestaltung der Grundlagen darf neben den vielen Forschungsprojekten nicht zu kurz kommen; sie verdient Unterstützung auch durch die Institutionen der Wissenschaftsförderung. Die Ergebnisse manifestieren sich hier nicht so sehr in Artikeln für Konferenzen, sondern eher in Büchern, über den Tag hinaus konzipiert sind, und auch in ausgefeilten Vorlesungen – zeitaufwändige Aktivitäten, die in der Regel weniger Anerkennung bringen als Projekte zu spezifischen Forschungsfragen von aktuellem Interesse.

Eine zeitgemäße Theorie ist in der Tat der Schlüssel für eine gute Lehre der Informatik. Wenn man sich mit einer verkrusteten oder flachen Theorieausbildung zufrieden gibt (und diese Tendenz gibt es), wird das Fach Informatik als Ganzes getroffen; zum Beispiel wird dann manche oder mancher für Informatik begabte (und daher anspruchsvolle) Studieninteressierte versucht sein, eher in benachbarten Fächern wie Mathematik, Physik oder Elektrotechnik an faszinierender Wissenschaft teilzuhaben.

Schlussbemerkung

Niemand, der in der Theoretischen Informatik arbeitet, kann sich der Faszination entziehen, die das immer enger vernetzte Spektrum an verschiedenartigen Zugängen mit sich bringt, sei es nun innerhalb der Theoretischen Informatik oder in der Kooperation mit den „Nachbarn“. Nur so scheint es auch möglich, die tiefgreifenden Herausforderungen anzugehen, denen sich die Theoretische Informatik gegenüber sieht. Die aktuelle Entwicklung der Theoretischen Informatik, wie sie beispielhaft in den Artikeln dieses Themenhefts dargestellt wird, macht klar, dass die Theoretische Informatik dabei auf einem guten Wege ist - als eine hochdynamische Disziplin, in der sich Bodenhaftung und glänzende Perspektiven verbinden.