Algebra und Algorithmik

Joachim von zur Gathen

Die beiden Gebiete im Titel sind schon über ihre Etymologie verbunden. Der berühmte Mathematiker und Astronom ↗ al-Khwārizmī schrieb im 9. Jahrhundert das Werk *al-kitāb al-muḫtaṣir fī ḥisābi al-jabri wā al-muqābalati (Das kurzgefasste Buch über Rechnen mit Ergänzen und Zusammenfassen von Ausdrücken)* über Algebra. Aus *al-jabr* ist die *Algebra* geworden, und der *Algorithmus* ist aus seinem Namen abgeleitet.

Die inhaltliche Verknüpfung der beiden Gebiete entsteht durch die Frage: wie kann man — möglichst effizient — die Aufgaben der Algebra algorithmisch implementieren? Die "Aufgaben" reichen von simpler Addition und Multiplikation hin zu Fragen der Algebrentheorie und algebraischen Geometrie. Die "Effizienz" hat stets mehrere Aspekte: einerseits möchte man immer bessere Algorithmen finden, andererseits kommt man manchmal hier nicht weiter und möchte dann beweisen, daß man bereits einen *optimalen* Algorithmus hat. Auf diesem Gebiet hat die *algebraische Komplexitätstheorie* schöne Erfolge vorzuweisen. Ein dritter Gesichtspunkt ist die technologische Umsetzung in den Systemen der ↗ Computeralgebra.

Ostrowski hat 1954 die Theorie begründet mit der Frage: ist ↗ Horners Regel zum Auswerten von Polynomen optimal? Das notwendige präzise Modell hat er mitgeliefert: das *nichtskalare Kostenmodell*, in dem man Multiplikationen und Divisionen zählt, aber Additionen und Multiplikationen mit Skalaren gratis sind. Die nichtskalare ↗ Komplexität eines Problems ist dann der minimale Aufwand von Algorithmen, die das Problem lösen. Die Auswertung von $\sum_{0\le i\le n} a_i x^i$ à la Horner kostet also n Operationen; hierbei sind $a_0, \ldots, a_n, x$ als Unbestimmte zu behandeln. Pan hat 1966 Ostrowskis Frage positiv beantwortet: es gibt keinen Algorithmus mit weniger als n nichtskalaren Schritten. Zu diesem Zweck hat er seine *Substitutionsmethode* eingeführt, die bis heute mit etlichen Variationen von Nutzen ist.

Die *Multiplikation* von Polynomen ist ganz billig in diesem Modell: man wertet die beiden Faktoren, deren Grad höchstens n sei, an $2n+1$ Stellen aus (hierzu muss der Grundbereich genügend viele Elemente haben), multipliziert die Werte und interpoliert. Die einzigen Kosten sind die $2n+1$ Multiplikationen. Anfang der 1970er Jahre haben dann Borodin und Moenck, Kung, Sieveking und Strassen gezeigt, daß sich auch die *Inversion modulo* x^n (mit ↗ Newton iteration) und die *Division mit Rest* in linearer Zeit erledigen lassen. Beim letzteren Problem sind $f, g \in F[x]$ gegeben mit Graden $n, m \ge 0$, und man sucht $q, r \in F[x]$ mit $f = qg + r$ und Grad $r < m$. Durch "Umdrehen" der Koeffizientenfolge entsteht etwa $\tilde f = x^n f(1/x)$, und es gilt $\tilde f = \tilde q \tilde g + x^{n-m+1}\tilde r \equiv \tilde q \tilde g \bmod x^{n-m+1}$. Hieraus ist $\tilde q$ durch Inversion von $\tilde g$ modulo x^{n-m+1} zu bestimmen, und dann q und r.

Des weiteren kann man den ↗ ggT von zwei Polynomen (und alle Quotienten im ↗ Euklidischen Algorithmus), die Werte eines Polynoms an n Stellen und, umgekehrt, die Interpolation, und die ↗ elementarsymmetrischen Funktionen mit $O(n \log n)$ Operationen ausrechnen. An der Entwicklung dieser Methoden waren Borodin, Brown, Fiduccia, Horowitz, Knuth, Moenck, Munro, Schönhage und Strassen beteiligt.

Diese *oberen Schranken* für die Komplexität gewinnen besonderes Interesse dadurch, daß Strassen entsprechende *untere Schranken* bewiesen hat. Mit anderen Worten: diese Algorithmen sind optimal (bis auf einen konstanten Faktor). Für seine *Gradmethode* betrachtet Strassen eine Berechnung mit l Schritten der Form $z \longleftarrow x \cdot y$ oder $z \longleftarrow x/y$, wobei x und y Linearkombinationen von Eingabevariabeln und früheren Zwischenresultaten sind. Jede solche Anweisung liefert eine quadratische Gleichung, und der Grad der zugehörigen ↗ affinen Varietät V ist nach ↗ Bézouts Ungleichung höchstens 2^l. Andererseits ist der Graph W der berechneten Funktion eine Projektion von V, so daß Grad $W \le 2^l$. Für die elementarsymmetrischen Funktionen etwa ist Grad $W = n!$, entsprechend den $n!$ möglichen Anordnungen der n verschiedenen Wurzeln eines Polynoms mit nichtverschwindender ↗ Diskriminante, und es folgt $l \ge \log_2 n! = \Omega(n \log n)$. Diese Schranke gilt auch für die Auswertung an vielen Stellen und für die Interpolation.

Baur und Strassen haben gezeigt, daß der Aufwand für die Berechnung sämtlicher partieller Ableitungen eines multivariaten Polynoms f höchstens das Dreifache der Kosten für f selbst ist. Hieraus folgt dann eine untere Schranke $\Omega(n \log n)$ für die "mittlere" elementarsymmetrische Funktion. All diese unteren Schranken gelten über einem beliebigen unendlichen Körper. Für endliche Körper hat Strassen ähnliche untere Schranken gezeigt.

In einem *Berechnungsbaum* führt man arithmetische Operationen aus und verzweigt je nach Ausgang eines Tests "$y = 0$?", wo y vorher berechnet wurde. Dies ist z. B. beim Euklidischen Algorithmus notwendig, wo der Grad eines Restes bei einer Division sich manchmal um mehr als 1 verringert. Strassen hat seine Gradmethode auch auf solche Probleme angewandt, und u. a. die Optimalität des schnellen Euklidischen Algorithmus von Knuth und Schönhage in einem starken Sinn gezeigt. Wenn nämlich n der Eingabegrad und $d_1, \ldots, d_l$ die Gradfolge der Quotienten ist, so kommt der Algorithmus mit $O(nh)$ Schritten aus, wo $h = H(d_1, \ldots, d_l)$ die ↗ Entropie bezeichnet. Und umgekehrt werden auch $\Omega(nh)$ Schritte

JOACHIM VON ZUR GATHEN (2000). Algebra und Algorithmik. In *Lexikon der Mathematik*, GUIDO WALZ, editor, 43–47. Spektrum Verlag, Heidelberg. ISBN 3-8274-0303-0.
URL http://cdroms.digibib.net/icafiles/LexMathe.ica.

benötigt für (↗ Zariski-) fastalle Eingaben, die die Gradfolge $d_1, \ldots, d_l$ liefern.

Über reellen Körpern ist es angebracht, dreifache Verzweigungen, entsprechend $<$, $=$ oder $>$, zu betrachten. Die resultierenden Berechnungsbäume entscheiden die Mitgliedschaft (des Eingabevektors der Länge n) in einer ↗ semi-algebraischen Menge $X \subseteq \mathbb{R}^n$. Nach Vorarbeiten von Steele und Yao hat Ben-Or 1983 gezeigt, daß für die Anzahl l von nichtskalaren Operationen und von Vergleichen gilt:

$$l \geq [\log(b(X) + b(\mathbb{R}^n \setminus X)) - n \log 3] \,/\, \log 6.$$

Hierbei ist $b(X)$ die Anzahl Zusammenhangskomponenten von X (in der üblichen Topologie). Der Beweis beruht auf der ↗ Milnor-Thom Schranke für b und der semialgebraischen Version des Morse-Sard Satzes. Es gibt mehrere Verallgemeinerungen, wo etwa b durch höhere ↗ Bettizahlen ersetzt wird.

Zu den hübschen Anwendungen zählt der Optimalitätsbeweis für $O(n \log n)$ Algorithmen für die ↗ konvexe Hülle H von gegebenen Punkten in der Ebene, und für die Suche nach einem größten Kreis mit Mittelpunkt in H, der keinen der Punkte im Inneren enthält.

Beim *Teilsummenproblem* sind Zahlen $a_1, \ldots, a_n, b$ gegeben und man fragt, ob es eine Teilmenge $S \subseteq \{1, \ldots, n\}$ gibt so, daß $\sum_{i \in S} a_i = b$. Mit ganzzahligen Eingaben ist dieses Problem $\mathcal{NP}$-vollständig. Überraschenderweise hat Meyer auf der Heide 1984 gezeigt, daß es Berechnungsbäume über $\mathbb{R}$ mit Grösse ungefähr n^5 für dieses Problem gibt. (Die Nichtuniformität dieser Bäume vereitelt den (vermutlich falschen) Schluß, daß das ganzzahlige Problem in $\mathcal{P}$ sei.) Andererseits erhält man mit Ben-Ors Methode, daß jeder solche Baum Grösse $\Omega(n^2)$ hat.

Solche nichtlinearen unteren Schranken sind eine besondere Stärke der algebraischen Komplexitätstheorie; entsprechende Resultate werden in der ↗ Booleschen Komplexitätstheorie vermutet, können aber bis heute nicht bewiesen werden.

Ostrowskis Modell, in dem skalare Operationen nicht gezählt werden, liefert asymptotisch gleiche obere und untere Schranken, ist aber nicht praxisgerecht. Das zentrale Problem ist die Multiplikation von Polynomen. Die beiden wichtigsten Algorithmen sind eine einfache Methode von Karazuba, mit $O(n^{\log_2 3})$ oder $O(n^{1.59})$ Operationen (wobei n der Grad der Polynome ist), und ein Algorithmus von Schönhage und Strassen, der auf der ↗ Schnellen Fouriertransformation beruht und nur $O(n \log n \operatorname{loglog} n)$ kostet. Die oben diskutierten Probleme können dann alle auch schnell gelöst werden; in den oben angegebenen Laufzeiten hat man jeweils n durch $n \log n \operatorname{loglog} n$ zu ersetzen, wobei jetzt alle arithmetischen Operationen gezählt werden.

Ein offenes Problem ist, in welchen Maschinenmodellen man etwa in Zeit $O(n \log n)$ multiplizieren kann — wie von Schönhage für *Random Access Maschinen* mit logarithmischen Kosten gezeigt — oder gar noch schneller, oder andererseits eine nichtlineare untere Schranke zu beweisen.

Die Multiplikation von Polynomen (modulo einem festen Polynom) oder von Matrizen gibt Beispiele von *bilinearen Abbildungen*. Wenn U, V und W endlich-dimensionale Vektorräume über einem Körper F sind und $f\colon U \times V \longrightarrow W$ bilinear, so bezeichnet $R(f)$ die bilineare Komplexität, also die kleinste Anzahl von Produkten zweier Linearformen — eine in U^* mal eine in V^* — deren Linearkombination f ist, mit Koeffizienten aus W. Dies ist gleich dem ↗ Rang des entsprechenden ↗ Tensors in $U^* \times V^* \times W$. Wenn $L(f)$ die übliche Komplexität von f bezeichnet, wie oben benutzt, so gilt $L(f) \leq R(f) \leq 2L(f)$.

Die Matrixmultiplikation $M_n\colon F^{n \times n} \times F^{n \times n} \longrightarrow F^{n \times n}$ (für $n \in \mathbb{N}$) spielt in der Entwicklung der bilinearen Komplexitätstheorie eine zentrale Rolle. Ein *erreichbarer Exponent* ist eine Zahl ω so, daß man M_n mit $O(n^\omega)$ Operationen berechnen kann. Der *Exponent* ω_0 ist das Infimum all dieser ω. Die Definition von M_n liefert den "klassischen" Algorithmus mit $\omega = 3$. Ein überraschendes Resultat von Strassen hat 1968 der gesamten Komplexitätstheorie einen wichtigen Impuls gegeben: es geht viel schneller! Er hat ein Schema für 2×2 Matrizen angegeben, das nur 7 (statt 8) Multiplikationen braucht. Indem er $n \times n$ Matrizen in 4 Blöcke mit halber Seitenlänge aufteilt, kann er das rekursiv anwenden und erhält $\omega = \log_2 7 < 2.81$.

Mit neuen Methoden wurden immer bessere Resultate erzielt; denkwürdig ist eine Oberwolfach-Tagung 1979, wo in einer Woche vier jeweils neue Exponenten gefunden wurden. Coppersmith und Winograd halten seit 1992 den Weltrekord: $\omega_0 < 2.376$.

Die bekannten unteren Schranken sind alle linear in der Eingabegrösse $2n^2$. Ein allgemeines Resultat von Alder und Strassen liefert $R(f) \geq 2 \dim A - \dim \operatorname{rad} A$ für den Multiplikationstensor f einer endlich-dimensionalen assoziativen Algebra A, also $R(M_n) \geq 2n^2 - 1$. Das beste Ergebnis ist Bläsers Schranke $R(M_n) \geq \frac{5}{2}n^2 - 3n$ von 1999.

Strassens allgemeine Theorie des Spektrums von bilinearen Abbildungen basiert auf zwei Operationen: der Tensorpotenz — entsprechend rekursiven Algorithmen – und der ↗ Degeneration — entsprechend einem Grenzübergang in der Zariskitopologie auf dem Raum der Tensoren.

Für die bisher besprochenen Aufgaben gab es offensichtliche Algorithmen, die ein in der Algorithmik grundlegendes Gütemerkmal besitzen: Laufzeit polynomial in der Eingabegröße. Dies ist

zunächst nicht der Fall bei anderen wichtigen Fragestellungen, etwa, nach aufsteigender Schwierigkeit geordnet: dem Faktorisieren von Polynomen, dem Lösen von polynomialen Gleichungssystemen und dem Entscheiden von algebraischen Theorien.

Beim Faktorisieren von Polynomen gibt es drei Grundaufgaben: Polynome in einer Variablen über endlichen Körpern und über den rationalen Zahlen, und die Reduktion von vielen auf eine Variable. Die erste Aufgabe hat Berlekamp Ende der 1960er Jahre gelöst, motiviert von der ↗ Kodierungstheorie. Die modernen Algorithmen, zu denen Cantor, Kaltofen, Shoup, Zassenhaus und der Autor beigetragen haben, können riesige Probleme lösen, etwa Zufallspolynome mit Größe von einem Megabit faktorisieren. Man vergleiche dies mit dem ↗ Faktorisieren von ganzen Zahlen, wo die Grenze des Machbaren (im Jahre 2000) bei etwa 500 Bits liegt. Die effizienten Algorithmen für Polynome benutzen interne Randomisierung; für die Theorie ist es ein offenes Problem, ob dies auch deterministisch (in Polynomialzeit) geht. Für Polynome über $\mathbb{Q}$ wurde von Zassenhaus das ↗ *Hensel Lifting* vorgeschlagen. Die ↗ Basisreduktion in ganzzahligen ↗ Gittern von Lenstra, Lenstra, Lovász liefert einen Polynomialzeitalgorithmus. Für multivariate Polynome teilt sich die Lösung in zwei Schritte: zunächst reduziert man von vielen auf zwei Variable, und dann — mit einer etwas anderen Methode — von zwei auf eine. Für den ersten Schritt benötigt man effektive Versionen von ↗ Hilberts Irreduzibilitätssatz, laut denen ein irreduzibles Polynom bei Substitution "im allgemeinen" irreduzibel bleibt. Hilberts Satz betrifft zum Beispiel die Substitution von $a \in \mathbb{Q}$ für t in $x^2 - t$. Hiervon kennt man bis heute keine Verschärfung, die effiziente Algorithmen liefert. Wenn man jedoch nur auf zwei Variable reduziert, also etwa in $x^2 - y^2 - t$ für t eine Linearkombination von x und y einsetzt, dann bleibt sogar bei zufällig gewählten Substitutionen die Irreduzibilität wahrscheinlich erhalten (Kaltofen und der Autor). Man erhält so auch dramatische Verbesserungen in den Abschätzungen für Emmy ↗ Noethers ↗ Irreduzibilitätsformen. Als Höhepunkt dieser Entwicklung liefern Kaltofens Methoden randomisierte Polynomialzeitalgorithmen zum Faktorisieren von multivariaten Polynomen im Modell der arithmetischen Schaltkreise und in dem der "black box" Darstellung.

Die algorithmische Frage nach einer effizienten Version des ↗ Fundamentalsatzes des Algebra, also das Approximieren mit beliebiger vorgegebener Präzision der reellen oder komplexen Nullstellen eines ganzzahligen Polynoms, ist naturgemäß stark numerisch orientiert. Ein effizienter Algorithmus von Schönhage braucht nicht viel mehr Zeit, als man zum Nachprüfen der Nullstelleneigenschaft benötigt.

Eine natürliche Verallgemeinerung des Polynomfaktorisierens ist die effiziente ↗ Wedderburnzerlegung von endlich-dimensionalen assoziativen Algebren. Über endlichen Körpern gibt es hierfür schnelle Algorithmen (von Eberly, Giesbrecht und anderen), während Rónyai gezeigt hat, daß sich das (vermutlich schwierige) Problem des Faktorisierens von quadratfreien ganzen Zahlen auf die Zerlegung von Algebren über $\mathbb{Q}$ reduzieren lässt.

Das Lösen von polynomialen Gleichungssystemen ist eine viel schwierigere Aufgabe. Matyasevichs Lösung von Hilberts 10. Problem hat gezeigt, daß dies über den ganzen Zahlen unentscheidbar ist. Man fragt daher nach reellen oder komplexen Lösungen bzw. Approximationen hieran. Die grundlegenden Methoden hierfür sind: die *zylindrische algebraische Zerlegung* von Collins, Buchbergers Algorithmus für ↗ *Gröbnerbasen* und die *charakteristischen Mengen* von Wu.

Mayr und seine Koautoren haben gezeigt, daß die Berechnung von Gröbnerbasen $\mathcal{EXPSPACE}$-vollständig ist. Dies bestätigt die praktische Erfahrung, daß Computeralgebrasysteme schon bei wenigen, etwa sechs oder acht, Variablen Mühe haben, eine Antwort zu finden. Der Frust wird gemildert dadurch, daß es bei vielen natürlichen geometrischen Problemen schneller geht, insbesondere wenn nur endlich viele Lösungen existieren. Die obere Schranke erhält man mit Hilfe der Theorie von paralleler linearer Algebra, in der man die üblichen Aufgaben für $n \times n$ Matrizen oder Gleichungssysteme in paralleler Zeit $O(\log^2 n)$ lösen kann (Berkowitz, Borodin, Chistov, Hopcroft, Mulmuley und der Autor).

Tarski hat 1949 einen Entscheidungsalgorithmus für die Theorie der reellen Zahlen angegeben. Dieser hat viele Verbesserungen erfahren; die momentan beste Abschätzung der Laufzeit ist doppelt exponentiell, wobei im obersten Exponenten die Anzahl von ↗ Quantorenalternationen (in ↗ Pränex-Normalform) steht. Ähnliches gilt für die Quantorenelimination über den komplexen Zahlen. Für diese Eliminationsprobleme gibt es auch entsprechende untere Schranken und ebenso für die Presburger-Arithmetik, die Theorie der natürlichen Zahlen unter der Addition (von Fischer und Rabin, Heintz und anderen). Die Theorien der endlichen Körper, die der endlichen Körper fester Charakteristik, und die der p-adischen Körper sind entscheidbar (Ax, Kochen, Ershov, P. J. Cohen), aber ihre genaue Komplexität ist unbekannt. Hingegen hat Kozen die Komplexität der Booleschen Algebra genau bestimmt.

Die wichtigsten Entwicklungen in der theoretischen Informatik bewegen sich um *Cooks Hypothese* "$\mathcal{P} \neq \mathcal{NP}$?", von Cook und Karp seit 1971 aufgeworfen. Valiant hat dies 1979 in die algebraische Domäne übertragen. Das Analogon

zu $\mathcal{P}$ bilden die *p-berechenbaren* Familien von multivariaten Polynomen, die mit polynomial vielen Operationen berechnet werden können. Dazu gehört etwa die Familie det $= (\det_n)_{n\in\mathbb{N}}$ der Determinante, wobei $\det_n$ die Determinante einer $n \times n$ Matrix mit n^2 unbestimmten Einträgen ist. Das Analogon zu $\mathcal{NP}$ bilden die *p-definierbaren* Familien $f = (f_n)_{n\in\mathbb{N}}$ von Polynomen, zu denen es eine p-berechenbare Familie g und eine polynomiale Funktion $t\colon \mathbb{N} \longrightarrow \mathbb{N}$ gibt so, daß $f_n(x_1, \ldots, x_n) = \sum_{e_{n+1},\ldots,e_{t(n)}\in\{0,1\}} g_n(x_1, \ldots, x_n, e_{n+1}, \ldots, e_{t(n)})$ für alle n ist. Valiants Reduktionsbegriff ist die *Projektion*, bei der man für eine Variable entweder eine Variable oder eine Konstante einsetzen darf. In der üblichen Weise erhält man dann den Begriff einer *p-vollständigen* Familie f: f ist p-definierbar, und jede p-definierbare Familie ist eine Projektion von f. Valiant hat gezeigt, daß die ↗ Permanente p-vollständig ist (bei von 2 verschiedener Charakteristik des Grundkörpers). Dies gilt auch für etliche Polynomfamilien, die gewisse kombinatorische Objekte abzählen, etwa die ↗ Hamiltonzykeln in Graphen. Das zentrale Problem hier ist es, *Valiants Hypothese* zu beweisen: es gibt p-definierbare Familien, die nicht p-berechenbar sind. Hierzu ist äquivalent, daß die Permanente nicht in polynomialer Zeit berechnet werden kann.

In der *algorithmischen Gruppentheorie* interessiert man sich u. a. für die effiziente Behandlung von ↗ Permutationsgruppen. Eine solche Untergruppe G der ↗ symmetrischen Gruppe S_n sei durch erzeugende Permutationen gegeben. Beim *Mitgliedschaftsproblem* ist ein weiteres $\sigma \in S_n$ gegeben, und man soll entscheiden, ob σ in G ist. Sims hat 1970 eine besonders nützliche Datenstruktur für G vorgeschlagen — die *starken Erzeugenden* — und Furst, Hopcroft und Luks haben dann das Problem 1980 in Polynomialzeit gelöst, und ebenso die Bestimmung der Gruppenordnung, der ↗ Auflösbarkeit und von normalen Abschlüssen. Weitergehende Arbeiten von Babai, Cooperman, Finkelstein, Kantor, Luks, Seress, Szemerédi und anderen haben verschiedene Aufgaben in Polynomialzeit gelöst, etwa die Bestimmung von ↗ Kompositionsketten. Die Richtigkeitsbeweise für einige dieser Algorithmen benutzen die ↗ Klassifikation der endlichen einfachen Gruppen. Andere Aufgaben, etwa den Durchschnitt zweier Untergruppen oder den ↗ Zentralisator eines Elements zu berechnen, scheinen schwieriger zu sein. Auf sie ist nämlich das Problem reduzierbar, ob zwei Graphen isomorph sind. Der Komplexitätsstatus des letzteren Problems ist unbekannt; wir wissen weder, ob es in $\mathcal{P}$, noch, ob es $\mathcal{NP}$-vollständig ist.

Eingehendere Beschreibungen und Literaturhinweise findet man in den Übersichtsartikeln [1, 2, 3] Präzise Aussagen und Beweise stehen im Standardwerk [4], und zu den Algorithmen auch in [5]. Die algorithmische Gruppentheorie ist ausführlich in [6] behandelt.

[1] Volker Strassen: Algebraic Complexity Theory, in *Handbook of Theoretical Computer Science*, vol. A, ed. J. van Leeuwen. Elsevier Science Publishers B.V., Amsterdam, und The MIT Press, Cambridge MA, 1990, 633–672.

[2] Volker Strassen: Algebraische Berechnungskomplexität, in *Perspectives in Mathematics, Anniversary of Oberwolfach 1984*. Birkhäuser Verlag Basel, 1984, 509–550.

[3] Joachim von zur Gathen: Algebraic complexity theory, *Annual Review of Computer Science* **3**. Annual Reviews Inc., Palo Alto CA, 1988, 317–347.

[4] P. Bürgisser, M. Clausen und M. A. Shokrollahi: *Algebraic Complexity Theory,* Grundlehren der mathematischen Wissenschaften **315**. Springer-Verlag, 1997.

[5] Joachim von zur Gathen und Jürgen Gerhard: *Modern Computer Algebra*. Cambridge University Press, 1999.

[6] Ákos Seress: *Permutation Group Algorithms*. Cambridge University Press, erscheint demnächst.