More parts than elements: how databases multiply

Adrian Mackenzie

a.mackenzie@lancaster.ac.uk

Cesagen, Lancaster University, UK

September 2010

wordcount = 9494 (including 1500 notes and references)

More parts than elements: how databases multiply

Abstract

Databases can be understood as the pre-eminent contemporary doing, organising, configuring and performing thing-and-people multiples in sets. Participation, belonging, inclusion and membership: many of the relations that make up the material-social life of people and things can be formally apprehended in terms of set-like multiples rendered as datasets. Mid-twentieth century database design derived different ways of gathering, sorting, ordering, and searching data from mathematical set theory. The dominant database architecture, the relational database management system (RDMS), can be seen as a specific technological enactment of the mathematics of set theory. More recent developments such as grids, clouds and other data-intensive architectures apprehend ever greater quantities of data. Arguably, in emerging data architectures, databases themselves are subsumed by even more intensive set-like operations. Whole databases undergo set-like transformations as they are aggregated, divided, filtered and sorted. At least at a theoretical level, the mathematics of set theory, as philosophically rendered by Alain Badiou, can also suggest some explanations of how multiples expand, ramify and split in databases. Badiou's account locates forms of instability or incoherence inherent in any set-based doing of multiples in the relation between parts and elements, between inclusion and belonging. Against the grain of Badiou's resolutely philosophical project, a set-theoretical account of databases might also point towards some vertiginous elements that elude regulation, norms and representation.

In July 2007, Google Corporation switched its principal service, Google Search, over to a re-indexed version of the web based on a different programming technique and database infrastructure called 'MapReduce.' That mostly undocumented switch to a slightly differently indexed web sent only mild tremors through the web (Shankland 2008). Google's MapReduce approach (Dean 2006), itself a topic of some controversy in the database design and architecture community, can be seen as a symptomatic response to the problems of managing rapidly expanding data sets.[1] Google's success as a business vitally depends on having a better response to this problem than other search engines. Architectures such as MapReduce allow large datasets to be disaggregated and re-aggregated more readily.

Why should a largely subterranean shift in how a large database – the Google web index – is made matter? Altering aggregations of data could be important for several reasons. First, aggregation in databases concerns the managing of multiples of peoples and things, and as such, should be of wider interest as deeply interwoven strands of experience. The socio-technical instantiation of many aspects of the contemporary world depend on database architectures and database management techniques. It is not coincidental that the most prominent contemporary trope for changes in the architectures of data processing, 'the Cloud' or 'cloud computing,' is both remote – clouds are above us – and relatively vaporous. As an infrastructure, 'the Cloud' does not involve earth moving or large construction projects. *The Economist* recently called 'cloud computing' the 'ultimate form of globalization' (Economist 2008) because it represents an expansive reconfiguration of data processing services that promises to transform how science, business, government and perhaps individuals, manage data. With the modesty and restraint typical of the IT industry, the CEO of Google Inc., Eric Schmidt claims that 'cloud computing is the story of our lifetime' (McDougall 2008). As Paul Jaeger argues (Jaeger 2009), cloud computing epitomises an attempt to de-localise aggregates of data, to re-distribute centres of calculation away from national, corporate, personal, business, military, government or scientific settings into 'dynamically scalable' data spaces. While there are many different motivations for the current shift to cloud computing (cost of hardware, elastic levels of demand, cheaper costs, meeting regulatory requirements, etc.), the key point for our purposes is that 'the Cloud' instantiates a symptomatic aggregation. Such aggregates increasingly rely on many concurrent processes that access selected parts of scattered domains of information, stored in highly organised yet flexible architectures.

Second, shifts in data aggregation pressurise – in several senses of the term – the boundaries between different commercial, scientific, technological, and regulatory domains. For instance, we are seeing the proliferation of databases and software for managing clinical trials of pharmaceuticals, connecting medical records and biomedical research databases. Clinical drug trials superimpose the constraints of regulatory approval processes, the logic of global pharmaceutical enterprises conducting international drug trials, the demands of clinical practice, the recruitment and organisation of cohorts of patients, the biopolitical statistics of epidemiology, as well as the organisation of high-throughput genomics research across institutional and commercial settings (Miller 2009; Davies 2009). A strong sign of the role of data management around health and medicine would be the move of large software, search engine and database companies such as Google, Microsoft and Oracle into the domain of health care. They regard the problem of aggregating scattered sources of personal health, information and medical records, as well as enrolling patients and patient groups in clinical trials of treatments and drugs as a major business opportunity. (Another symptom would be the extraordinary recent growth of business analytics in association with the web and internet (Davenport and Harris 2007; Ayres 2007; Rosenberger and Nash 2009).) At the very least, these boundary shifts deserve scrutiny.

Third, contemporary large-scale data architectures can be seen as an index of nascent post-liberal worlds. 'Vertical aggregates,' it has been argued, are transforming the 'lateral aggregates' of globalised worlds (Papadopoulos 2008).[2] Vertical aggregates superimpose economic, state, political and cultural processes in shifting consortia of local, regional, national, transnational and international actors. Symptoms of vertical aggregation can be seen in factories, offices, clinics, airports, media, offices, homes and cities. Any encounter with vertical aggregates requires some way of thinking about aggregation, about how things that lack both full unity and complete dispersion take shape. Vertical processes of aggregation occurring in post-liberal settings challenge social theory to re-think relations between one and many, between universality and plurality. Aggregates of interest to social theory such as 'population,' 'corporation,' 'State,' 'consortium,' 'human resources,' or 'community' are often practically entwined with databases. While Dimitris Papadopoulos and Vassilis Tsiano's exemplar of a vertical aggregate is a new BMW automobile factory designed by Zaha Hadid in Leipzig, Germany (Papadopoulos and Tsianos 2007, 150), contemporary data architectures such as Google's Map-Reduce could also be an exemplar of vertical aggregation. Databases have been a crucial component in the flattening of processes of production, exchange, consumption and communication in lateral globalisation. They have also undergone constant modifications. We might say that the superimposition of data in vertical aggregation occurs in a tectonic process like subduction. In geology, subduction describes what happens at the boundary between two continental plates as one is forced under the other. Here I use subduction as a tectonic metaphor, but also in a slightly different, more generic sense to refer to a process of subtraction, removal or withdrawal that occurs when many elements – geographical, legal, scientific, economic, commercial, institutional, private and public data – are constantly forced into different intersecting sub-sets that form new parts. Importantly, via Badiou, I hope to show that this forcing down or subduction cannot be completed.

History of multiples: from R to MapReduce

Database architectures are relatively invisible and intricate socio-technical forms. In cultural and media theory (Manovich's work on the database form and media (Manovich 2001)), in surveillance studies (Lyon 2007; Lyon 2003), and in science studies (Bowker 2005), database forms have achieved some recognition. There is also much work on the historical, political and philosophical status of number and calculation that forms a background to any account of databases (for instance, (Elden 2006; Rotman 1993; DeLanda 2002; Porter 1996; Desrosieres 1998). However, there is very little work that addresses the historical, ontological status of multiples presented in the architecture and formalisms of database architectures as they emerged in the mid-twentieth century (Neufeld and Cornog 1986). This is somewhat surprising given the recognised centrality of databases in information societies, network cultures, etc., and the generally acknowledged shift in new economy and new media to 'database-driven' processes. There are perhaps just too many important examples to list here, but any such list would include Web2.0 or social networking sites such as Facebook or MySpace, customer relations management (CRM), business-to-business (B2B) processes, national security awareness, scientific models, supply chain management, market analytics, bioinformatics, and perhaps, the epicentre of all contemporary searching and sorting tasks, the production indexing system for Google's web search engine.

For a database-oriented social theory, one critical decision concerns where to cut in analysing databases as historically sited encounters with the multiple. I cut across the diversity of database applications, products, techniques and problems by treating all databases as attempts to work on multiples, and more specifically, treating them as multiples inflected by twentieth century mathematics of set theory. What happens if we examine databases as set-based doing of multiples? In asking this, there is a risk of falling into some form of mathematical idealism. It is worth taking that risk because it allows us to draw on the philosopher Alain Badiou's engagement with set theory (Badiou 2000; Badiou 2004; Badiou 2008). Over the last forty years or so, Badiou has worked with the mathematics of sets as primary resource for thinking multiples (Badiou 2000; Badiou 2004; Badiou 2008). In particular, Badiou treats set theory as an ontology of the multiple.[3] His work might offer some ways of thinking about the ontological texture of multiples and relations in databases.

A trenchantly ontological account of multiples (as distinct from the multiplicities that have attracted so much attention in Deleuzean theory) such as Badiou's begins by identifying numbers as multiples: the 'Being of number is the pure form of the multiple' (Badiou 2008, 58). Unless we take number to be a-historical, this statement does not idealise multiples. Just the opposite: if number is the pure form of the multiple, then changes in numbering, counting and sorting will affect doing of multiples. Multiples are historical becomings. That is, multiples – numbers, sets, groups – are not the same thing at all times and places. They differ in degree and kind. This assertion can be supported from various perspectives. Science studies scholars such as Helen Verran (Verran 2001), Anne-Marie Mol (Mol 2003) and John Law

(Law 2004) argue for the need to develop ways of thinking between the one and the many. They seek to sharpen analytical focus on how multiples are devised and situated. An ethno-mathematical-influenced account such as Helen Verran's *An African Logic* shows how counting itself depends on embodied relations, and concludes:

> What I am suggesting is that there are multiple ways to do the relation unity/plurality; hence there are multiple sorts of numbers. (Verran 2001, 107)

Whatever approach is taken – philosophical, sociological or anthropological – the status of a multiple today is difficult to conceive apart from the technical processes of ordering, sorting, counting and calculating. Multiples are often done today in computing, data-processing, modelling and visualisation. Data multiples in many domains – scientific, government, commercial, media, etc. – nearly always pass through databases. Multiples are increasingly mediated through databases and database-driven architectures. While multiples in the world remain innumerable, databases have increasingly emerged as a way of collecting, enumerating and enunciating multiples in particular ways. The existence of vast databases (such as Google or Amazon or GenBank) and the existence of huge numbers of databases (including many personal databases that would not be visible or obvious as such to the people who rely on them to manage their gadget-equipped lives), I would argue, can be read as a ramification of particular historical renderings of multiples. Databases are a situation in which we materially encounter the doing of the multiple. They embody historical forms of multiple that imbue spaces, movements, communication, governmentalities, knowledges and values with techno-ontological textures.

Making connections between database logics and set multiples might allow us to think more broadly about databases as encounters with the multiple. There are some signs of submergence of relational database management systems (RDMS) as the standard form of the database. Today, the 'relational model of data for large shared banks' first developed by Edgar F. Codd at IBM under the name 'R' in the late 1960s (Codd 1970) is being engulfed by vastly simplified yet expansive data models such as MapReduce. This shift away from the relational database can be seen as a result of the weight of peta-bytes and exa-bytes of data generated by billions of people using electronic devices and network technologies such as the Web. If RDMS enables lateral aggregates, MapReduce – as well as being an important factor in Google's ongoing dominance of the web-search service market (Powell 2009) – might embody some of the tendencies to vertically aggregate. The evolutionary explosion of databases from the RDMS of the early 1970s through to 'MapReduce' and cloud computing of 2009 can be understood as a multiplication of multiples. This shift can also be regarded as an opportunity to think about the multiple in different ways, and to look for ways in which multiples constantly exceed their embodiments in particular technical systems of management. Certain threads of Badiou's mathematical ontology – the concepts of forcing, generic subset, and evental site – offers a way of conceptualising the multiplying of multiples.

From tables to tuples

Relational database management systems (RDMS) became conceivable only in the wake of the mathematics of set theory. Key features of RDMS are predicated on set-like multiples. The central structure of almost any contemporary commercial database architecture (and many other data aggregates that would not be immediately obvious as databases) is the *table.* Tables are omni-present in relational databases. For instance, one of the key constructs in the industry standard SQL – Structured Query Language – is 'CREATE TABLE', and every query on a relational database eventually concerns tables. How have tables – rather than for instance, hierarchical tree structures – prevailed as the primary form of the multiple? In some ways, the success of tables as ways of spatially organising and working with information comes as no surprise. Histories of information science and organisation tell of the long-lived practices of table-making dating from 2500 BCE Sumerian tax and harvest tables through astronomical tables, mathematical tables, statistical and actuarial tables, timetables to the spreadsheets of the late twentieth century (Campbell-Kelly 2003). Tables comprise sets of elements in a simple spatial relation comprising vertical and horizontal lines, columns and rows. Each column lists elements of a defined type. Each row connects a set of defined types. For instance, each row of a database table might include a person's name, an age, gender and a residential address. The flatness of the table, its capacity to be recorded and displayed across flat surfaces, is both limiting and enabling.

The relational databases first commercialised in the 1970s by IBM and Relational Databases Inc. (now Oracle Corporation) transformed table-making practices through a mathematical re-structuring derived from set theory and relational algebra. An important mathematical constructs runs across tables in DBMS. The mathematical notion of 'relation' is used to model a great number of concepts such as 'is equal to,' 'is greater than,' 'is adjacent to,' 'is congruent to,' etc. The sheer variety of relations that can be modelled partly explains the extraordinary adaptability of the relational database to different settings. Edgar F. Codd's work on relational databases is indicative of this intensification. In his original article, Codd states:

> The term *relation* is used here in its accepted mathematical sense. Given sets S_1, S_2, • • • , S_n (not necessarily distinct), *R* is a relation on these *n* sets if it is a set of *n*-tuples each of which has its first element from S_1, its second element from S_2, and so on (Codd 1970, 379).

This slightly opaque rendering of the notion of relation conveys something of the mathematical relation first described by the 19th century British mathematician, Augustus De

Morgan in somewhat philosophical terms: 'when two objects, qualities, classes, or attributes, viewed together by the mind, are seen under some connexion, that connexion is called a relation'. It is very important to recognise that a relation readily spans many sets. Any particular relation selects a trajectory through a field of possible relations.[4] The 'Cartesian product' suggests something of the multiplication of relations across tables (the sets). A database design selects a subset from the many possible connections between tables. Such subsets subtract the fixity of tables and replace it with a more generic relationality, that of the *tuple*. Relations allow tables to be merged, split, aggregated, sorted and searched in 'schemas.' Database 'schemas,' often comprising dozens of tables, each with several columns and many rows, frame a mutable set of connexions. Hence the relation *R* that Codd brought to bear on databases is a set of sets, since a tuple (typically embodied in a 'row' in a database table) itself is a set that expresses some order.

The very idea of a relation as a set of tuples seems stunningly simple, yet, as is often the case, the layering, convolution and multiplication of relations generates new dynamics. While Codd was not the first to make use of sets and set theory as a way of thinking about data management (for instance, several years earlier, a paper on a set-theoretical understanding of databases had appeared (Childs 1968)), the relational algebra he brought to bear on the problems of data organisation and storage proved powerful. As Codd wrote, 'since relations are sets, all of the usual set operations are applicable to them' (Codd 1970, 383). The 'usual set operations' referred to here include unions, intersections, complements and 'products' (associating every element of one set with every element of another'). Through these operations, many new relations can be derived from the base relations found in a database. In treating all relations as sets and as subject to set operations, the flat form of the table undergoes a process of subduction. It is pushed down into a much more turbid zone where boundaries and distances become more fluid.

Database designers sometime perform a process of 'normalization' or 'elimination of non-simple domains' (Codd 1970, 381). Normalization (a term that Codd putatively borrowed from a television broadcast of a speech by President Nixon on 'normalizing' U.S. relations with China) dismantles and dis-aggregates many existing tables in order to clear the way for creation of more widely spanning relations. The upshot of the normalization process is constructive: the potential to generate many relations by melding disparate things, by defining relations that observe certain constraints, and by remaining open to production of more relations, always in the simple domain of tuples. The flatness of the normalized form remains indifferent to the difference between sets or domains, to their hierarchical or convoluted intersections and unions. Like many aspects of relational database architecture, normalization is often done not only with a view to the performance of the database (database performance issues such as speed and churn attract a cluster of technical specialists and a sub-industry of services and products), but with ongoing change in mind.

Another important SQL database construct, 'CREATE VIEW,' derives new relations within and across domains without altering the underlying structure of the database:

> Views serve many purposes including increased security (by hiding attributes from applications and/or users without a legitimate need for access) and enhanced performance (by materializing views defined by complex SQL queries over very large input tables). But views are primarily used to allow old programs to operate correctly even as the underlying database is reorganized and redesigned (Gray et al. 2005, 5).

Views allow a database to appear to remain the same, even as underlying architectures expand and change. Finally, perhaps the definitive and relentlessly repeated set-like operation on a relational database is 'SELECT FROM … WHERE.' Any query to a database takes the form of a SELECT command. The syntax of SELECT ranges from extremely simple requests for a single row of a single table to highly complex intersections, unions, and joins spanning many tables. The combinatorial power of the SELECT statement across relational databases can be seen today on Web2.0 and smart-phones as they generate suggestions, connections, menus, recommendations and invitations, or manage large aggregate identities and groups, or concatenate vast numbers of links, images, files and user information. Because SELECT makes subsets or parts of the sets totally available, Web2.0 can seem to be highly-responsive or 'user-produced.'

Sets in action: open-closed, inclusion, belonging and non-cohesion

Given that relational databases instantiate contemporary doing of multiples, how do the underlying formalisms of set theory help us apprehend them differently? What leverage can the mathematical formalisation of set-making offer to social theory? Here, I offer only three observations derived from Alain Badiou's set-based account of multiples: (1) the opposition between closed and open is not the most relevant distinction to make in sorting multiplies; (2) rather, there are different modes of being-in sets – belonging and inclusion – and these different modes of being-in sets have different implications; (3) the irreducibility of different modes of being-in sets generates non-cohesion, dynamism, and excess, and this irreducibility generates movements and events. While these lines of thinking are abstract in some ways (Badiou's writing, I think it could be said, embodies an energetic drive to philosophical abstraction), such abstractions actually lead quite concrete lives in many settings. In a world where database abstractions entwine around and inflect many levels of experience (from standing in a check-in queue at an airport to looking for things on supermarket shelves: see

(Baker 2008) for a popular account of data management and mining practices around buying, selling, advertising, security, and health), we need multiple ways of comprehending flows and structures of data, accompanying them as such, and at the same time, seeing where and when they change, collapse or expand.

(1) Between open and closed

The operations that structure and alter databases – 'create table,' 'views from' or 'select' – all derive directly from set-theoretical operations – union, complement, cartesian product, intersection, complement – on relations. In that respect, the underlying form of the database multiple is the set. The cumulative effect, however, of billions or trillions of set operations on multiples remains unclear. While contemporary databases are said to be vast and expanding, is there any qualitative change or change in kind possible here?

Most critical understandings of what might happen in contemporary data-spaces focus on opposition between open and closed multiples. In social theory and philosophical thought more generally, open forms of belonging are affirmed over closed forms (in notions of public space, public sphere, democracy and freedom in much political theory, in ideas of property and technology, in Martin Heidegger's notion of the history of being as Opening, in Giorgio Agamben's 'the Open' as definitive of the animal-human difference, in the conceptualisation of multiplicity in Gilles Deleuze's work, etc.). Open seems to imply inclusion, whereas closed implies exclusion. Asymmetries and differences in power, value, ownership or authority derive from closure. When the power of databases to govern or regulate movement (for instance, in the intensified border control systems of the U.S.A. and Europe), or to supplement and commodify identity (for instance, in the quasi-automated recommendation and targeting advertising systems of many Web2.0 sites) comes into discussion, the opposition between open and closed often underpins normative stances. So, for instance, surveillance studies has evaluated databases from the standpoint of who controls access to information, and how databases undermine citizen or consumer rights and freedoms. On the one hand, database are seen as either threatening to enclose situations that should remain open (e.g. freedom to move without restraint). On the other hand, databases make visible relations or attributes of individual or groups that should remain more or closed (e.g. the privacy of personal habits, financial or health records).

While these concerns are not trifling or invalid, something else is at stake here, something that exceeds the opposition of closed and open that drives many of the debates. It is possible that any staging of the space of databases in terms of open vs. closed will only occlude the different ways databases configure multiples. Badiou provocatively argues that the distinction between open and closed is dispensable:

> The set is the exemplary instance of something that is thinkable only if one dispenses entirely with the opposition between the closed and the open. ... We could even say that the set is that neutral-multiple which is originarily subtracted from both openness and closure, but which is also capable of sustaining their opposition (Badiou 2004, 73).

Why should the set only be thinkable if we dispense with the opposition between open and closed? (What Badiou means by 'a neutral multiple … originarily subtracted from both openness and closure' will be discussed below.) In part, Badiou can draw support from mathematics: the opposition between open and closed does not play an important role in set theory. Set theoretical definitions of open and closed sets do not make a decisive distinction between them. An open set, as Badiou points out in his own philosophical rendering of the mathematical definitions, is 'substantially established once we possess a multiple such that the we dwell within it by taking the intersection of two elements, or the union of as many elements as we wish (even an infinity of elements)' (Badiou 2004, 73). We can see from this definition that the relational database actually takes openness as its principle of operation. Openness – to the insertion of new elements, to the derivation of new relations, to new orderings and structurings, to 'the union of as many elements as we wish' – is the basis of the relational architecture as a set-based doing of multiples. This openness allows databases to aggregate on sometimes global scales.

(2) Including more than belonging

If from the perspective of set theory, open vs. closed is somewhat derivative in making sense of multiples, then what differences do matter? A more important difference concerns how things get into in sets, the difference between belonging and inclusion, between elements and parts. Again drawing directly on the results of set theory and wrapping those results in a philosophical language, Badiou writes:

> there are always more parts than elements; i.e. the representative multiplicity is always of a higher power than the presentative multiplicity. This question is really that of power (Badiou 2004, 157).

This is a key set-derived desideratum in Badiou's thought. Subsets are *included* in sets as parts, while elements *belong* to sets. We have already glimpsed some signs of this in the theory of the relational database as advanced by Codd. The 'VIEW' construct in SQL, for instance, applies this difference. Discrete elements of data that belong to a table remain untouched by subsequent overlays of different subsets or parts of the database. When they normalize a database, database designers or architects work with this distinction, and in fact, attempt to make it stronger. Normalization does not change the elements that belong to the database, it alters the number of parts included in it. Many debates around data-subjects, data-doubles, data-identity would perhaps benefit from a clearer grasp of the distinction between inclusion and belonging, a distinction that runs underneath and across any differences between open and closed. No-one belongs to a database as an element, but many aspects of contemporary lives are included as parts of databases.

(3) A generic subset subtracted from every predicate

Badiou insists that the difference between an element in a set and a part of a set is highly generative: 'the immanent excess that ‘animates’ a set, and which makes it such that the multiple is internally marked by the undecidable, results directly from the fact that it possesses not only elements, but also parts' *(Badiou 2004, 78)*. How does this immanent excess, the linchpin of Badiou's philosophical work on the multiple, play out in set theory, and how does it play out in database practice? Badiou traces the excess of parts over elements, and of inclusion over belonging, from the late nineteenth century work of Georg Cantor (Cantor 1915) on set cardinality (countability) to the 1960s work of Paul Cohen (Cohen 1966) on the unprovability of the 'continuum hypothesis,' an important and ongoing problem in set theory. While I am skirting around the technical steps between Cantor's and Cohen's work that Badiou himself takes in order to arrive at the excess of inclusion over belonging, the accepted position in contemporary mathematics seem to align with Badiou's outcome.[5] Specifically, the construction of the 'generic subset' using the technique of 'forcing' developed by Cohen lies at the heart of this demonstration of excess of parts (or subsets) over elements. The careful mathematical construction of the generic subset so that it remains constitutionally incomplete and escapes full predication is a mathematical accomplishment for which Paul Cohen received the most prestigious prize in mathematics, the Fields Medal, in 1966. Describing the implications of Cohen's work, Badiou writes:

> You certainly cannot straightforwardly name the elements of a generic subset, since the latter is at once incomplete in its infinite composition and subtracted from every predicate which would directly identify it in language. But you can maintain that is *if* such and such an element *will have been* in the supposedly complete generic subset, *then* such and such a statement, rationally connectible to the element in question, is, or rather will have been, correct. Cohen describes this method – … – as that of *forcing' (Badiou 2004, 130)*.

Again, there are many difficulties in reading this kind of theoretical statement since Badiou draws so heavily on mid-twentieth century set theory – a terrain that is scarcely familiar to most readers in social sciences or humanities – to stage arguments around truth, experience, politics or aesthetics. The main difficulty in understanding Badiou on this point stems from the rather complicated idea of a generic subset. Much in Badiou's work pivots on the concept of the generic subset. While the notion of a subset is not hard to understand, the concept of a generic subset is much more complicated since it brings in several different set-related set concepts ('countable antichain,' 'dense subset,' 'set filter,' etc.). The generic subset embodies the impossibility of complete predication, classification or grouping of a multiple. As Badiou suggests, 'the generic subset, we might say, contains a little bit of everything, so that no predicate ever collects together all its terms. The generic subset is subtracted from predication by excess' (Badiou 2004, 110). The generic set then is a slightly paradoxical entity: by virtue of excess, it is missing or substracted.

From flat to vertical: the forces of subduction

How do we get from this mathematical-ontological scene of excess-subtraction back to the database? Supposing we provisionally accept the philosophical conceivability of this unnameable set-derived subtraction in even the most strictly defined, flattened, and thoroughly predicated multiples, does the forcing that makes a generic set have any implications for thinking about the sociality of database forms? Badiou himself displays no particularly interest in addressing socio-technical practice. Yet the subtraction of that which exceeds predication, and hence the excess of inclusion over belonging, must affect in some way how people do and are done by the contemporary multiples embodied in databases. Although the affected processes would need to be worked out in much more detail, we can locate the forcing that produces a generic subset and the excess of inclusion over belonging at a more tangible level by turning to contemporary changes in the architectures of large-scale data centres (Gray et al. 2005).

It is often said that data doubles every three years (Chen and Schlosser 2008, 1). Scientific and commercial data centres cannot rely on existing relational database architectures to manage growth, partly because it is hard to predict exactly what data will be needed when or where, but also because the relations implicit to such data have become vastly more implicit, conjunctive and extensive. Extensive accumulations of data across many domains of communication, retail, government, commerce, entertainment, healthcare and various sciences all suggest the possibility of predicating ever more inclusive aggregates. Perhaps the very elements themselves as listed in the table of the database contain parts or subsets that for certain purposes need to be disaggregated. Instead, the growth of business analytics (see (Baker 2008) for a popular account) in many domains derives answers to problems by cross-hatching the relational data space with new modes of inclusion. Everything should relate in principle to everything else, if only the right way to annotate, merge, or agglomerate datasets can be found.

The implication of the generic subset, and the forcing that produces it might be this: although larger and faster relational databases are constantly being built for commercial, government, military and scientific uses (Oracle 2009), the excess that animates the multiple might not exactly retreat in the face of scaling up. The lesson of the generic subset might be that more schemas, views, clusters and tables also multiply the multiple in various ways. In a sense, everything we saw concerning relations and the relational algebra of the RDMS tended to assume the possibility of predicating differences that matter in the form of tables, views and relations, in other words, in subsets. Database architects collect and group predicates in meaningful or measurable subsets. The relational databases, the query languages with their elaborate syntax and the management techniques all seek to make differences predicable; that is, the subject of a qualitative or quantitative predicate. Yet this belief in predication, in finding the full set of relations that connect all relevant elements, is troubled by subductive process, a forcing under of relations that makes predication and validation every more exorbitant.

This exorbitant predication appears in various ways. In many contemporary domains the implication of many relations in a single element is a mundane yet ongoing problem.[6] Let us return to the re-indexed Google web search based on MapReduce. MapReduce might be read as an attempt to predicate differences in ever quicker succession by reducing search-times. Yet it also acknowledges the impossibility of grasping or sorting them fully. MapReduce envisages constant growth in bioinformatics, the web, business analytics, and environmental sensing data. We have seen how set-theoretic actions allowed relational databases to protect users of data from the details of how data is stored and searched. Programmers still had to have some understanding of the elements and parts of a dataset in order to construct useful queries and updates on the database, but how the database sorted and searched data was largely only a concern for software companies like Oracle or Microsoft or academic computer scientists researching database architectures. Map-Reduce also tries to protect programmers from having to know where and when data is stored or sorted, but now the emphasis falls on a constantly shifting infrastructure of servers, storage and network connection. The database architectures that previously supported infinitely variable set-making possibilities are subducted into a moving architecture that itself can be constantly re-aggregated in set-like unions and intersections.

How does MapReduce do this? Again, MapReduce is very generic. It can be applied to many different settings. The two key researchers, Jeffrey Dean and Sanjay Ghemawat write in describing the work done at Google:

> MapReduce is a programming model and an associated implementation for processing and generating large data sets that is amenable to a broad variety of real-world tasks (Dean and Ghemawat 2008, 107).

The programming model can be used in many different ways by programmers, but it basically requires them to express problems in terms of two functions, 'Map' and 'Reduce.' The idea is that many of the things that people want to find in data can be more found more quickly if the data is parcelled into separate bundles and searched in parallel. For instance, in order to keep its web search engine up to date, and therefore maintain its hold on users and advertisers, Google Corporation computers constantly crawl the web looking for new or changed web pages. Naturally, this process generates data. Much of this data can be searched and sorted in parallel. Any application of the MapReduce model assumes that elements of data have known relations to other elements, so they can be parceled into many parts (the 'Map' step), processed, and then later re-combined (the 'Reduce' step). In order to do this parceling, the design of a program needs to be reorganized according to the 'stunningly simple model' (Lämmel 2007, 1) of MapReduce: 'the user of the MapReduce library expresses the computation as two functions: map and reduce' (Dean and Ghemawat 2008, 107). These functions have been borrowed from LISP, the functional programming language, itself derived from the mathematical work of Alonzo Church on 'lambda calculus' (Church 1996). 'Map' partitions a domain of elements into parts that can be processed separately to produce intermediate results, and 'Reduce' takes the intermediate results and unifies them into a final result, effectively, a new part. Current 'implementations' include the Amazon Elastic Compute service, Google's own data centres, Yahoo's Cloud, and a host of other 'cloud computing' initiatives (see (Economist 2008) for a survey of commercial ones).[7]

MapReduce offers a way of organising data-processing databases into new parts. Databases themselves undergo predication as they are transformed into parts. Yet these parts change the very texture of multiples. MapReduce subducts set logic into the data management infrastructure. This pushing down of set logic into infrastructure, as Dean and Ghemawat write, 'hides the messy details of parallelization, fault tolerance, data distribution and load balancing' (Dean and Ghemawat 2008, 107). At the same time, this very pragmatic solution to the problem of writing code to partition large collections of records, tables, files, documents, web pages and other data creates meta-structure or vertically aggregated sub-sets. Rather than replacing existing databases, MapReduce subducts them in a multi-tiered meta-structure.

From the perspective of the unnameable excess crystallised in the techniques of forcing and the generic subset, what are we to make of MapReduce? MapReduce is a symptom of the kind of 'forcing' that seeks to deal with non-cohesion between different modes of being in sets. We might view MapReduce as an attempt to render a certain kind of data processing infrastructure subject to set-like predication. Badiou describes such meta-structures as the 'state' of a situation:

> The state is a sort of metastructure that exercises the power of the count over all the sub-sets of the situation. Every situation has a state. Every situation is the presentation of itself, of what composes it, of what belongs to it. But it is also given as state of the situation, that is, as the internal configuration of its parts or sub-sets, and therefore as re-presentation (Badiou 2004, 157).

Stretching Badiou's formulation, MapReduce is an 'internal configuration' that attempts to render the 'state' of the contemporary data situation with its unstable mixtures of inclusion and belonging. It re-counts the situation – the proliferation of data multiples – as a configuration of parts or subs-sets. It presents the space of data-centres in ways that afford new subjectifications (the post-genomic scientist, the Web2.0 user, the self-fashioning database-literate patient groups), new forms of value (the recommendation services, the personalised, targeted advertising, the personalized genotyping and health profiling) and new configurations of public-private space (in security and control, in mapping and logistics provision, in manufacture, distribution and media).

Data multiples as evental sites

A 2009 article in *The New York Times* described how U.S. university students were set the challenge of using MapReduce-based clusters of computers to solve problems, not because these problems need solving but in order to ensure that they acquire requisite skills in working at with vast datasets (Vance 2009) Here the process of subjectification is framed as an economic imperative to train students for encounters with large-scale data multiples. At the same time that it generates a state, or a meta-structure that aggregates a situation in order to re-present a configuration of subsets, MapReduce might become an evental site, which for Badiou represents an up welling of fresh political power. This admittedly is a much more tenuous possibility since at the moment it is hard to see anything in this situation that would match the requirements for an event according to Badiou's criteria: 'There is', writes Badiou, 'an event only in so far as there exists a site for it within an effectively deployed situation (a multiple)' (Badiou 2004, 101). A situation can only generate events via a site. A site allows events to occur because it affects how elements belong to it. Again Badiou relies on the slippage between inclusion and belonging to stage his account of evental site:

> One could say there is a sort of 'fragility' peculiar to the site, which disposes it to be in some sense 'wrested' from the situation. This fragility can be formulated mathematically: the elements of an evental site are such that none of their own elements belong to the site (Badiou 2004, 101).

While it makes sense to think of the site of an event as a fragility prone to subtraction or subduction in a situation, a domain in which greater instability or uncertainty allows things to happen, what is to be gained by enunciating this fragility mathematically?

On the one hand, Badiou's whole set-based thinking points to the impossibility of controlling or fully determining the immanent, discordant excess of multiples. In this setting, we can read Badiou's work as offering a way to posit the necessary possibility of an experience of this excess without reduction. As he puts it, 'it is certainly the case that experience must, each and every time, re-determine this immanent excess' (Badiou 2004, 79). If we accept Badiou's conviction that thinking multiples today means engaging with number mathematically, then transformations in databases too 're-determine' the immanent excess. One could imagine this happening in many different ways, ranging from all the work-arounds, techniques, tricks and indeed habits and enculturation that allow us to readily sort, group, collect and retrieve things and relations.

On the other hand, there are certain techniques and processes that expose aspects of situations more amenable to events. The primary of these is 'truth.' Truth, in politics, art, love and science requires an evental site, somewhere to take place. Truth is a kind of event because it wrests something from a situation that does not fully belong to it. Here is Badiou's version of that process, quoted here at length because it leads straight into set-making processes:

> The act of the subject is essentially finite, as is the presentation of indiscernables in its being. Nevertheless, the verifying trajectory goes on, investing the situation through successive indifferences. Little by little, what takes shape behind these acts begins to delineate the contour of a subset of the situation – or of the universe wherein the evental axiom verifies its effect. This subset is clearly infinite and remains beyond the reach of its completion. Nevertheless, it is possible to state that if it is completed, it will ineluctably be a generic subset (Badiou 2004, 116).

Much in this formulation is not too unfamiliar – it is easy to accept that complete truth is not available; that subjects are finite; etc. However, the identification of the 'act of the subject' in verifying something in a situation with a rigorously defined incomplete subset, 'a generic subset,' is much stronger. To say that truth, if it were available, would take a particular form that is comprehensively defined as intrinsically incomplete, a generic subset, is neither idealist or empiricist, realist or constructivist, contextual or non-contextual, but all of these at the same time. A generic subset, as we have seen, is a way of precisely constructing a form through excess something of ineradicably missing. If truth were a generic subset, then the subject who experiences truth, whose experience is *of* verification, would also be exposed to this incompleteness. Almost everything we have seen regarded databases seems to head in the opposite direction, towards reducing excess to productive, predicated ends.

Conclusion: multiples include the void

That databases embody contemporary sensibilities of moving, gathering, counting, sorting, and ordering multiples is hardly debatable. What does the mathematics of set theory supply to database-embodied multiples today beyond a logical foundation for database theory? Coupled with the philosophical effort to articulate mathematics as ontology, the mathematical orientation puts into question the assumption that we already know how multiples move in and around databases, how they are textured, how they mix openness and closure, how they wrest inclusion from belonging. The 'relation' as theorised in set theory and practiced in relational databases differs from relations in social theory more generally. In databases, a relation is a way of presenting a multiple by selecting elements from it according to a set of constraints ('select … from … where ….'). This might provoke for social scientists different ways of thinking about parts and elements in their own work.

The dis-aggregation and re-aggregation we see in Map-Reduce subducts multiples. 'Subduction' or forcing, rather than completing multiples, seems to multiply them. Yet the ways in which multiples are made in databases is not just expansive (as in lateral aggregation). Data does not simply expand to predicate more and more actions, events or phenomena. Modes of inclusion change alongside modes of belonging. The elements in a database (for instance, the 'required fields' demanded by so many web forms) do not map directly onto the parts of the database or the subsets that may be formed from the dataset as it is sorted and searched. Indeed while new kinds of elements may occasionally be added to a database schema, the re-partitioning and selection of subsets seem far more important in trajectories through databases. Contemporary shifts from relational databases to MapReduce data architectures are usually narrated in terms of the pressures of expanding data, or as an attempt to capture what is still missing. Vertical aggregation brings different ways of making sets. Nascent multiples have a different texture, temporality and logic than lateral aggregates. On this reading, the history of database architectures is one of subduction: relational databases subject to MapReduce processing become more vertically aggregated. Vertical aggregates contour different subsets, parts, situations, sites and states.

Yet, this aggregation is driven by something missing. It is this incompleteness, the constitutive incompleteness crystallised in the generic set, that Badiou's set-based thinking of the multiple seeks to enunciate. Badiou speaks of what might happen at the 'edge of the void':

> I determine [events] ontologically (with all the required mathematical formulations) as that which is “on the edge of the void” – that is to say, that which is *almost* withdrawn from the situation's regulation by an immanent norm, or its state. In a situation (in a set), it is like a point of exile where *it is possible* that something, finally, might happen (Badiou 2000, 85).

'On the edge of the void' – here 'void' is understood in set-theoretical terms as the name for the empty set – is in ontological terms a name for 'there is' (Badiou 2004, 84). To be 'on the edge of the void' means to be part of what in a given situation cannot be fully mathematized, fully regulated by norms or state.

This kind of breakdown in set-making is not necessarily welcomed by database designers or architects, even as they are constantly driven to act on it. It might take many forms ranging from negotiations over what fields are to be included in a table, or how a thing is to be coded in some registry or inventory, to dimly felt senses of incomplete inclusion in datasets experienced by people. Given this, the set-theoretical ontology Badiou develops might also help us to think differently about parts and elements, inclusion and belonging for people, things, events and processes. Can we, as social scientists, find something in this subducted or 'almost withdrawn' possibility – the excess of inclusion over belonging – that would allow us to re-construct the phenomenological texture of living data aggregates? What in our accounts of multiples would be different if the incompleteness of our own knowing, our predication, was understood in terms of the forcing that makes a generic set? By definition, it is difficult to name something concrete since the subduction of multiples re-configures existing forms of concreteness, existing sensibilities of relationality, grouping, membership and belonging.

1 MapReduce has been widely discussed and implemented. The most popular implementation is an open source project called 'Hadoop' released by the Apache Software Foundation, and used by FaceBook, Amazon, AOL and Yahoo amongst many others (Apache Software Foundation 2009).

2 The social consists of vertical aggregates containing and intermingling segments of social classes, social subjectivities, or other social groups into large formations along an imagined commonality. These social bodies condense economic, technoscientific, political, and cultural power, and control decision-making processes. They are different from the social structures we have known up to this moment. (Papadopoulos and Tsianos 2007, 153)

3 While a substantial secondary philosophical literature has developed around Badiou's work (Hallward 2004; Hewlett 2007; Barker 2002), there is little or perhaps no work available that tries to bring his mode of philosophical thought to bear on scientific, technological or commercial settings. This paper attempts to do that.

4 More technically, a relation L over the sets $X_1, \ldots, X_k$ is a subset of their Cartesian product, written $L \subseteq X_1 \times \ldots \times X_k$.

5 For a readable account, see (Chow 2007)

6 One example would be bioinformatics and its treatment of biological data. Homology search or more generically, 'sequence search' is a typical contemporary data problem in that it deals with large volumes of sequence data generated by increasingly high-throughput automated genome sequencing projects. Life scientists face many problems in making meaningful use of such data. While the data is relatively homogeneous (for instance, a long sequence of DNA base pair 'GCATGCCCAAA'), the relations running through that data are deeply latent. A DNA sequence cannot be automatically divided into useful elements and stored in a database. In order to begin to render different elements of a sequence more salient, one thing biologists often do is to compare a new sequence with existing sequences. If the new sequence is homologous with an existing sequence that has been already analysed and annotated in some database, then the new sequence may have similar biological function, it may indicate evolutionary affinity, or it may simply identify what organism the sequence belongs to. Searching for a homology or good alignment between two long given sequences is usually computationally challenging, but when a sequence needs to be compared to the many sequences found in databases such as GenBank, computational burden increases substantially. The multiple here is full of elements whose relation to each other is almost purely 'ordinal' (that is, they come first, 23rd, 10034th, etc). In post-genomic sciences, I would suggest, we have a case where problems of excess multiples, of countability, and of connectability between relata, and above all, of what subsets are included, become very salient. We might say that sequence homology, a key enabling technical process in biopower and biocapital (Rose 2006; Rajan 2006; Helmreich 2007) embodies the increasingly 'infinite composition' of multiples today. As sequences multiply (e.g. in personalized genomics) and grows longer, the algorithmic complexity needed to compare them grows exponentially. An extended account of homology searching or sequence alignment is beyond the scope of this article. One account of the process and its algorithmic architecture can be found in (Mackenzie 2006) See Chapter 3).

7 What 'real world' tasks are amenable to MapReduce? A patent held by the Web search engine service Yahoo Inc. applies MapReduce to the bioinformatics problem of homology search. The patent, entitled ' Bioinformatics computation using a MapReduce-configured computing system' (Hsiao, Dasdan, and Yang 2008) claims that the homology search for large cross-genome analysis can be made into something like a websearch using MapReduce. Rather than submitting a chosen sequence to an online service and then waiting for a hours or days until results come back, a user would be able to carry out homology search at the same rate as they do web searches; that is, within seconds.

References

Apache Software Foundation. 2009. Welcome to Apache Hadoop! July 16. http://hadoop.apache.org/.

Ayres, Ian. 2007. *Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart*. 1st ed. Bantam, August 28.

Badiou, Alain. 2000. *Deleuze: The Clamor of Being*. Theory out of bounds v. 16. Minneapolis: University of Minnesota Press.

———. 2004. *Theoretical writings*. Ed. Ray Brassier and Alberto Toscano. London ; New York: Continuum.

———. 2008. *Number and Numbers*. Cambridge: Polity.

Baker, Stephen. 2008. *The Numerati: In Which They'll Get My Number and Yours*. First Edition. Jonathan Cape Ltd, November 6.

Barker, Jason. 2002. *Alain Badiou: A Critical Introduction*. London: Pluto.

Bowker, G. C. 2005. *Memory practices in the sciences*. MIT Press Cambridge, MA.

Campbell-Kelly, Martin. 2003. *The history of mathematical tables*. Oxford University Press.

Chen, S, and SW Schlosser. 2008. *Map-Reduce Meets Wider Varieties of Applications*. IRP-TR-08-05, Technical Report, Intel Research Pittsburgh.

Childs, David L. 1968. Description of a set-theoretic data structure. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 557-564. San Francisco, California: ACM. doi:10.1145/1476589.1476663. http://portal.acm.org/citation.cfm?id=1476663.

Chow, Timothy Y. 2007. A beginner's guide to forcing. *0712.1320* (December 9). http://arxiv.org/abs/0712.1320.

Church, Alonzo. 1996. *Introduction to mathematical logic*. Princeton N.J.: Princeton University Press.

Codd, E. F. 1970. A relational model of data for large shared data banks. *Commun. ACM* 13, no. 6: 377-387. doi:10.1145/362384.362685.

Davenport, Thomas H., and Jeanne G. Harris. 2007. *Competing on Analytics: The New Science of Winning*. 1st ed. Harvard Business School Press, March 6.

Davies, Kevin. 2009. Jay Tanenbaum Urges Collaboration To Treat the Long Tail of Disease. *Bio-IT World*. February 26. http://www.bio-itworld.com/2009/02/26/tenenbaum-mmtc-keynote.html.

Dean, Jeffrey. 2006. Experiences with MapReduce, an abstraction for large-scale computation. In .

Dean, Jeffrey, and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, no. 1: 107-113. doi:10.1145/1327452.1327492.

DeLanda, Manuel. 2002. *Intensive Science and Virtual Philosophy*. London & New York: Continuum.

Desrosieres, Alain. 1998. *The Politics of Large Numbers: A History of Statistical Reasoning*. Cambridge, Mass: Harvard University Press.

Economist. 2008. A survey of corporate IT: : Computers without borders. *The Economist*, October 23. http://www.economist.com/specialreports/displaystory.cfm?story_id=12411854.

Elden, Stuart. 2006. *Speaking against number: Heidegger, language and the politics of calculation*. Edinburgh: Edinburgh University Press.

Gray, J, DT Liu, M Nieto-Santisteban, A Szalay, DJ DeWitt, and G Heber. 2005. Scientific data management in the coming decade. *ACM SIGMOD Record* 34, no. 4: 34-41.

Hallward, Peter. 2004. *Think again : Alain Badiou and the future of philosophy*. London ; New York: Continuum.

Hewlett, Nick. 2007. *Badiou, Balibar, Rancière: Re-Thinking Emancipation*. London: Continuum.

Hsiao, Ruey-Lung, Ali Dasdan, and Hung-Chih Yang. 2008. BIOINFORMATICS COMPUTATION USING A MAPRREDUCE-CONFIGURED COMPUTING SYSTEM. June 5. http://www.patentlens.net/patentlens/patsearch.cgi?patnum=US_2008_0133474_A1.

Jaeger, Paul T. 2009. Where is the cloud? Geography, economics, environment, and jurisdiction in cloud computing. *First Monday* 14, no. 5. http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/2456/2171.

Lämmel, R. 2007. Google’s MapReduce programming model—Revisited. *Science of Computer Programming* 68, no. 3: 208-237.

Law, John. 2004. *After method : mess in social science research*. London ; New York: Routledge.

Lyon, David. 2003. *Surveillance as Social Sorting: Privacy, Risk, and Digital Discrimination*. London: Routledge.

———. 2007. *Surveillance Studies: An Overview*. Cambridge: Polity.

Mackenzie, Adrian. 2006. *Cutting code: software and sociality*. Digital Formations. New York: Peter Lang.

Manovich, Lev. 2001. *The Language of New Media*. Cambridge, MA: MIT Press.

McDougall, Paul. 2008. Google, IBM Join Forces To Dominate 'Cloud Computing' -- Network Computing -- InformationWeek. *InformationWeek: the Business Value of Technolgoy*. May 1. http://www.informationweek.com/news/services/data/showArticle.jhtml?articleID=207404265.

Miller, Rich. 2009. Pharmaceuticals Test-Drive the Cloud « Data Center Knowledge. *Data Center Knowledge*. May 26. http://www.datacenterknowledge.com/archives/2009/05/26/pharmaceuticals-test-drive-the-cloud/.

Mol, Annemarie. 2003. *The Body Multiple: Ontology in Medical Practice*. Durham, N.C: Duke University Press.

Neufeld, M. Lynne, and Martha Cornog. 1986. Database history: From dinosaurs to compact discs. *Journal of the American Society for Information Science* 37, no. 4 (7): 183-190. doi:10.1002/(SICI)1097-4571(198607)37:4<183::AID-ASI2>3.0.CO;2-W.

Oracle. 2009. Oracle 11g, Siebel, PeopleSoft | Oracle, The World's Largest Business Software Company. http://www.oracle.com/index.html.

Papadopoulos, Dimitris. 2008. *Escape routes : control and subversion in the twenty-first century*. London ;;Ann Arbor MI: Pluto Press.

Papadopoulos, Dimitris, and Vassilis Tsianos. 2007. How to Do Sovereignty without People? The Subjectless Condition of Postliberal Power. *boundary 2* 34, no. 1 (Spring2007): 135-172. doi:Article.

Porter, T. M. 1996. *Trust in numbers: The pursuit of objectivity in science and public life*. Princeton Univ Pr.

Powell, James. 2009. Q&A: Harnessing Computing Power for Data Analysis with MapReduce --

Enterprise Systems. *Enterprise Systems*. http://esj.com/Articles/2009/06/09/Data-Analysis-MapReduce.aspx?p=1.

Rosenberger, Larry E., and John Nash. 2009. *The Deciding Factor: The Power of Analytics to Make Every Decision a Winner*. Jossey-Bass, March 16.

Rotman, Brian. 1993. *Ad Infinitum: The Ghost in Turing's Machine: Taking God Out of Mathematics and Putting the Body Back in: An Essay in Corporeal Semiotics*. Stanford U.P.

Shankland, Stephen. 2008. Google spotlights data center inner workings | News Blog - CNET News. *cnet news*. May 30. http://news.cnet.com/8301-10784_3-9955184-7.html.

Vance, Ashlee. 2009. Training to Climb an Everest of Digital Data. *The New York Times*, October 12, sec. Technology. http://www.nytimes.com/2009/10/12/technology/12data.html?_r=2&th&emc=th.

Verran, Helen. 2001. *Science and An African Logic*. Chicago, London: The University of Chicago Press.