THE SCIENTIFIC WORKS OF RAINER KEMP (1949–2004)

PHILIPPE FLAJOLET, MARKUS NEBEL, AND HELMUT PRODINGER‡

Abstract. This short note presents a summary of the scientific contributions of Rainer Kemp (1949–2004) in the area of discrete mathematics, combinatorial enumeration, and analysis of algorithms. A complete bibliography of Kemp's publications is included.

Rainer Kemp, Professor of computer science at the Johann Wolfgang Goethe University of Frankfurt, passed away on May 14, 2004, eventually yielding to an illness that he had been fighting for one full year. His research career had started more than thirty years earlier: he defended his thesis supervised by Günter Hotz in 1973 and, already in 1972, he counted amongst the very few contributors to the first pan–European conference on theoretical computer science, the by now famous *ICALP*. In 32 years of academic scientific activity, Kemp wrote altogether 53 publications, of which a complete list appears in the bibliography section of this note.

Kemp's research started with problems in the theory of formal languages, a *by then* prevalent branch of computer science in Europe. He was especially interested in "syntax analysis" or parsing and, even after switching subjects, would return to formal languages every so often. However, soon after his beginnings, his interests started to drift towards discrete mathematics, especially combinatorial enumeration and asymptotic methods, with many of the problems he considered being motivated by analysis of algorithms. It is in this area that he published the vast majority of his papers. At the same time, Kemp played an important rôle in organizing, jointly with the two senior authors of this note (P.F. and H.P.) the first three meetings held at Schloß Dagstuhl in 1993, 1995, and 1997. These meetings to which he dedicated much of his energies would play a crucial rôle in shaping up a community now referred to as *"AofA"*, a nickname for *Analysis of Algorithms.* The AofA community could accordingly develop a strong base in Europe as well as numerous ramifications worldwide, and we all are greatly indebted to Kemp for his commitment.

We analyse below some of Kemp's papers and have chosen to organise the presentation of his works into ten categories, to which is added a separate section on his book published by Teubner–Wiley in 1984. These works all bear Kemp's mark: they are systematic, thorough, sometimes even extreme in their calculational aspects. They are testimonies of Kemp's meticulous and rigorous attitude to scientific activity. As we all know, technical papers invariably contain minor errors and misprints—even a glance at the collected papers of a purist like G.H. Hardy will confirm this fact. In the case of Kemp, we are not aware of more than a handful such things in the whole of his works.

Kemp's sense of accuracy and completeness may render his works difficult to access by the uninitiated. This should not hide the fact that they contain a number of gems

Date: February 6, 2005; revised April 6, 2005.

‡This material is based upon work supported by the National Research Foundation of South Africa under grant number 2053748.

based on deep insights. At a time where several chapters of computer science are drifting towards a "quick–and–easy" publication attitude, many of Kemp's works can serve as model as what a carefully thought scientific production should be. Also, many of them contain highly original ideas, whose power would only be revealed in later years—Kemp in many ways was a precursor, though he was never to become, by his own choosing, a school builder. For instance, in the first work of Kemp that we discuss below, the one relative to register allocation, he was the very first to recognize explicitly the importance of the Mellin transform. Kemp's work and lectures served as an eye–opener: nowadays, Mellin related techniques have become a standard apparatus of the average–case and probabilistic analysis of algorithms, and probably about a hundred papers in the area appeal to it in some way. (P.F. remembers his first encounter with Kemp around 1979. Kemp: "*This is a Mellin transform!*". Flajolet: "*Oh! What is this?*". Then Kemp went on to explain. . .)

Leftist trees is another area where Kemp made spectacular advances. In his first edition of *The Art of Computer Programming*, Volume 3, published in 1973, Knuth had an innocuous looking "exercise":

> **5.2.4.34**. [*M47*] How many leftist trees with N nodes are possible, ignoring the `KEY` values? [The sequence begins 1, 1, 2, 4, 8, 17, 38, . . . ; is there a simple asymptotic formula?]

Kemp solved that one brilliantly fourteen years after it had first been posed as an open problem. It is worth noting that the ranking "[*M47*]" places this problem amongst perhaps a dozen of comparable difficulty throughout the 2,000 odd pages of Knuth's *magnum opus*. It appears that only three or four of that level of difficulty have been solved so far (e.g., sorting networks, the binary gcd algorithm, the two–stacks problem). Despite his usual soberness and modesty, Kemp is definitely to be remembered for that feat.

Much of Kemp's work was otherwise dedicated to the analysis of major parameters of trees. Trees of algorithmic theory can be broadly classified into three groups: the "square–root" trees (related to combinatorial tree models as well as to branching processes), the "logarithmic" trees (associated to order structures), and the digital trees or "tries" (based on the digital structure of data). Kemps' work was almost entirely dedicated to the first type, which is also the one most relevant to formal languages, automata, and parsing.

Kemp will also be remembered for having been the first writer ever to have dedicated a book solely to the analysis of algorithms. He was of course preceded by Knuth, but the purpose of Knuth's books is largely the classification and formalisation of fundamental algorithms and data structures, with analytic techniques being only one of the aspects considered. Kemp's example was to be followed by a series of other works by later researchers so that, in this area, we now have treatises by Hofri [G], Mahmoud [K, L], Sedgewick–Flajolet [P], and Szpankowski [R].

Rainer Kemp's ideas will survive, not only by the strength of his major results, by his articles published in many of the primary journals of computer science, but also by the presence of his book on the shelves of many university departments.

1. Register function and arithmetic Expressions

That was a well–known open problem that ended in a tied race between Flajolet, Raoult, Vuillemin [F], and Kemp [5]. If a binary tree is used to represent an arithmetic expression, what is the optimal (i.e., minimal) number of additional registers to evaluate the tree? The strategy is to always evaluate the more difficult subtree first—this was discovered by Ershov and published in the very first volume of *Communications of the ACM* [E]. In this way, there is a recursive labeling of the nodes of the tree. Kemp was able to show that the average of this parameter, provided that all binary trees with n internal nodes are equally likely, admits an explicit form, namely

$$2+\frac{n+1}{\binom{2n}{n}}\sum_{k\geq 1}\bigl(1+v_2(k)\bigr)\left[\binom{2n}{n+1-4k}-2\binom{2n}{n-4k}+\binom{2n}{n-1-4k}\right],$$

where $v_2(k)$ is the number of trailing zeros in the binary representation of k.

From this, using approximations for binomial coefficients and the Mellin transform (in the version presented in Knuth's Volume 3) allowed Kemp to evaluate this quantity asymptotically, leading to $\log_4 n+\delta(\log_4 n)+o(1)$, where $\delta(x)$ is a periodic function with explicitly given Fourier coefficients. Kemp's discovery of oscillations in an asymptotic expansion arising from combinatorics and analysis of algorithms was anticipated by works of Knuth (based on suggestions by De Bruijn). However, Knuth's fluctuations are relative to models that involve a superposition of basic patterns at different scales, in fact, powers of 2. (This is presented in the first edition of Volume I of *The Art of Computer Programming*, where a Mellin-like technique is applied to digital trees and radix-exchange sort; see p. 131 of [I].) Kemp's oscillations are quite different in that their amplitude is "visible" (as opposed to a mere 10^{-5} in the case of radix-exchange sort). More importantly, they would appear to be related to fluctuations present in many number representation systems. We now have a fair understanding of these subtle phenomena. Kemp's work served to draw attention to an interesting area of research that lies somewhere between asymptotic analysis and basic analytic number theory.

A second strategy for evaluating a binary tree (the corresponding arithmetic expression) traverses the tree in postorder, using a stack to store intermediate results. Here, the stack size (the left–height) of the tree is equal to the maximum number of cells of the stack used by the algorithm. The research of Kemp related to this parameter will be discussed in Section 3.

In [17] Kemp invented a different class of algorithms for the evaluation of arithmetic expressions represented by binary trees. Each algorithm makes use of an input–restricted deque of length k and an auxiliary storage. These algorithms can be considered as generalization of the stack based strategy mentioned before. Kemp proved that the number of binary trees with n leaves which can be processed by means of a deque of length k and an auxiliary storage of size i is equal to the number of all trees with n leaves which can be processed by means of a stack of maximum length $(k+1)(i+1)-1$. The corresponding sets are different. Additionally, paper [17] contains a detailed average case analysis of the space complexity of the new class of algorithms.

Kemp's paper together with the parallel one by Flajolet, Raoult, and Vuillemin presented the first analysis of register allocation in the average case. In fact, Kemp's paper contains, only slightly hidden, a complete analysis of the associated probability distribution. It was to be followed by works of several authors including Kirschenhofer and Prodinger [H]. It is worthy of note that the register function is truly an important structural parameter of trees. In this capacity, it is to be encountered in the statistical study of river networks, being well known to students in hydrogeology under the name of "Horton–Strahler number." It has been otherwise encountered by Yekutieli and Mandelbrot [T] in the study of self–similar branching systems, where phenomena analogous to those first uncovered by Kemp were later found to surface. It has been used in image synthesis, where "real" trees (the ones we see everyday) are to be modelled and graphically rendered by computer programs. It is especially pleasant to observe in this case that a fundamental question raised by the nascent science of computing turned out to have resonances in so many seemingly unrelated areas of research. Kemp's research on the subject places him as an important link in this chain.

2. Leftist trees

Kemp solved an open problem from Knuth in [24]: the enumeration of leftist trees. A *leftist tree* is a binary tree such that the distance from any node x to any of the leaves in the subtree having x as the root is minimal for its leftmost leaf. Denote by t_n the number of leftist trees with n internal nodes and the generating function

$$H(z) := \sum_{n\geq 1} t_n z^n.$$

Kemp derived the implicit equation

$$H(z) = z + \frac{1}{2}H^2(z) + \frac{1}{2}\sum_{k\geq 1} T_k^2(z)$$

with

$$T_1(z) = z, \quad T_2(z) = zH(z), \quad T_{k+1}(z) = T_k(z)\Big[H(z) - \sum_{1\leq i<k} T_i(z)\Big], \ k \geq 2.$$

From this, he was able to show, by appealing to the Darboux–Pólya method of complex asymptotic analysis, that around $\eta = 0.363704\ldots$, $H(z)$ behaves like a square root:

$$H(z) = H(\eta) + \sum_{k\geq 1} b_k(\eta - z)^{k/2}.$$

Consequently,

$$t_n = [z^n]H(z) \sim 0.250363\ldots(2.749487\ldots)^n n^{-3/2}.$$

In a further paper [30] Kemp went on and considered the average of several additive parameters of leftist trees. Examples are *left/right branch length, (free) external path length, left/right path length.* In [47], the "leftist" concept was applied to 2–3–trees (as opposed to binary trees, as before). Surprisingly, the number of these objects is counted by Catalan numbers, and thus eventually related to binary trees. Accordingly, Kemp finds a bijection between leftist 2–3–trees and binary trees, each with n nodes. In [46, 48], the concept has been generalized further to *simply generated trees.* The latter is an important and very general class of trees introduced by Meir and Moon in the mid 1970s [N]. In [49], the previous correspondence "leftist 2–3–trees versus binary trees" has then been lifted to a great level of generality. Proofs are both analytic and bijective.

As already signalled in the introduction, Kemp's solution to Knuth's problem was in its time a *tour de force*. His work is nowadays to be regarded as a component belonging to a general stream of ideas. The goal is to extract asymptotic information from combinatorial counting sequences in cases where no explicit expression is available neither for the sequence itself nor for its generating function. The works of Kemp in this area can serve as a link between Pólya's original work [O] on the enumeration of chemical isomers and present–day research whose aim is to categorize functional equations arising from combinatorics and relate them to asymptotic–probabilistic properties.

3. Stack size and Height

The stack size of a binary tree can be easily understood as follows: For each path from the root to the leaves, count only the left edges. (The terminology is adequate since this parameter measures the memory cost incurred by a traversal, which is achieved by a stack.) The stack size is then the maximum of these numbers. Equivalently, if one considers the corresponding planar tree (via the rotation correspondence), then stack size transforms into height.

Such a tree is translated into a Dyck path (non–negative lattice path). In [11], the following question is addressed: what is the level of this path, after t units of time, if $t = 2\rho n$, and a constant $0 < \rho < 1$. The average level is found to be proportional to

$$4\sqrt{\tfrac{\rho(1-\rho)}{\pi}}\sqrt{n},$$

and higher moments as well as explicit formulæ are also obtained. The paper [8] is an earlier version of that, which additionally contains an extension of the classical paper by de Bruijn, Knuth and Rice [D] about the height of planar trees (=stack size of binary trees), in terms of higher moments and the distribution function (given by an elliptic Θ-function).

In [14], this study is continued. Kemp is interested in local maxima ("max–turns") resp. minima ("min–turns") in planar trees. He shows that the j–th max/min–turn is asymptotic to $8\sqrt{\frac{j}{2\pi}} \pm 1$, for fixed j and large n. More refined results are available.

Dyck paths related to trees are "closed" (in the sense that they return to zero after $2n$ steps). In [12] Kemp considers open paths (non–negative); the level at time $2n$ is *not* specified. He is again interested in the height (maximal level) of these objects. Result: the average is asymptotic to $(\ln 2)\sqrt{2\pi n}$.

In [9], planar trees with root degree r (fixed) are considered. The height of general planar trees was determined in the above mentioned paper by de Bruijn, Knuth and Rice. Now Kemp considers the height of this subclass. This is a major computational effort, and the average is determined to be asymptotic to $\sqrt{\pi n}+1-\frac{r}{2}$, so the parameter r occurs only in the second order term. In [16], he fixes another parameter: the number m of leaves. It is natural to let m grow with n, like $m = \rho n$, with $0 < \rho < 1$. Result: the average height is asymptotic to $\sqrt{\pi(\rho^{-1} - 1)}\sqrt{n}$.

In [32], the study of the stack size of binary trees was continued. The value of the stack size occurs at the root, when one labels the nodes recursively according to a simple rule. Now information about nodes with certain values attached is derived, in particular the number of those nodes (called "critical" nodes) where both subtrees are equally difficult (in terms of stack size). Surprisingly enough, there is a proportion of $\pi^2/6 - 1$ of critical nodes, asymptotically.

The paper [7] also studies the (average) stack size of trees, but this time the trees are the derivation trees generated by linear context–free grammar. Such grammars are reasonably manageable (not too different from regular languages), and lead, in terms of generating functions, to matrices, so that the Perron–Frobenius machinery can be applied. Consequently, the average stack size is always of the form $nF(n)+G(n)+$s.o.t., with periodic functions $F(n)$ and $G(n)$.

Kemp's line of research started, as already said, from the analytic treatment of the analysis height of Dyck paths by de Bruijn, Knuth, and Rice. Kemp showed that in fact the methods could be extended, sometimes at the expense of considerable technical efforts, and then the estimates could be greatly refined. Some of his results were extended by Louchard [J], who dealt with multiplicatively weighted paths—these are of immediate relevance to a wide class of "amortized" (average–case) complexity analyses. Louchard succeeded in developing a useful probabilistic intuition, based on Brownian motion and Gaussian processes, which provides an easier access to several asymptotic phenomena of the type earlier considered by Kemp, whenever only dominant and subdominant regimes are sought.

4. Formal Languages and Enumeration

Let L be a formal language over an alphabet X. The quantity

$$d(L) := \lim_{n\to\infty} \frac{\text{number of words of length} \le n \text{ in } L}{1 + |X| + \cdots + |X|^n}$$

is called the asymptotic density of L, provided it exists. It was known, largely from the works of Berstel in the early 1970's [C], that regular languages have rational asymptotic densities, while unambiguous context–free languages have algebraic ones. However, for an inherently ambiguous context–free language, the situation was unclear. Kemp [10] constructed the first example of a context–free language with a non–algebraic density. The main ingredient here is the series

$$F(z) = \sum_{\lambda\ge 0} z^{-2^\lambda}$$

(a so–called "Fredholm series"). The function $F(z)$ is known to be transcendental for z being a natural number > 1.

In [15], Kemp considered the set $S = \{w \in \Sigma^* \mid w = w^R\}$ of all palindromes over an alphabet Σ. In particular, he was interested in the number of words of length n in S^2. To cite one result, here is the generating function of these numbers:

$$1 + \frac{1}{4}\sqrt{|\Sigma|}\sum_{j\ge 1}\varphi^{-1}(j)z^j\left[\frac{\left(1+\sqrt{|\Sigma|}\,\right)^2}{\left(1-\sqrt{|\Sigma|z^j}\right)^2} - \frac{\left(1-\sqrt{|\Sigma|}\,\right)^2}{\left(1+\sqrt{|\Sigma|z^j}\right)^2}\right],$$

with the function φ^{-1} (related to Euler's totient function) defined by

$$\varphi^{-1}(n) = \prod_{p|n,\ p \text{ prime}} (1-p).$$

Möbius inversion plays an important rôle here. The asymptotic density of S^2 can also be determined (it is zero for alphabets of size at least 2).

Kemp's contributions in this area are quite spectacular. He solved a very natural conjecture in formal language theory, in a line of problems launched some ten years earlier by Berstel, Schützenberger and others. The connection between formal generation mechanisms and the associate density phenomena has witnessed some revived interest recently. In this area, Kemp was undeniably a pioneer. From a methodological point of view, it is interesting to observe the way certain conjectures in formal languages could find solutions by way of number theory and mathematical analysis.

5. Other combinatorial papers

The paper [31] appeared eight years after it was conceived; it was somehow lost in the jungle of the editorial office–perhaps the journal's editors should figure in the Guiness Book of Records for this feat. Kemp considers ordered (=planar) trees: Let $B_{n,k,r}$ be the number of all trees with n nodes, root–degree r and height $\leq k$, and $Q_{n,k,r}$ the number of trees with n nodes, height k and exactly r nodes achieving that height (having a distance k from the root and no children), then

$$Q_{n,k,r} = B_{n+1,k,r+1} - B_{n+1,k,r} + B_{n,k,r-1}$$

(later, Strehl found a bijective proof of this [Q]). Kemp derives explicit albeit very complicated expressions for these quantities. On an asymptotic level, he shows that the probability that a tree has a height attained r times approaches 2^{-r} as $n \to \infty$. Moments of this distribution are also evaluated.

In [29], Kemp constructs a bijection between:

— t–ary trees with $tk+1$ nodes and stack size $s+1$ and
— ordered trees with $(t-1)k+1$ nodes, allowed node degrees $d \equiv 0 \bmod (t-1)$ and stack size s.

A somewhat similar paper is [35]. Here, the bijection is between

— t–ary trees with $tk+1$ nodes and stack size $< 2t$ and
— t–ary trees with $tk+1$ nodes and maximal height.

In [37], Kemp considers 0–balanced ordered trees (=all leaves are on the same level): the number of these with n nodes, m leaves, and height h is given by

$$[t^{n-m-h+1}]\big(1+t+\cdots+t^{h-2}\big)^{m-1}.$$

The analysis goes along the lines of runs of ones in binary strings. The average height of such trees with n nodes is asymptotically $\sim \log_2 n + \mathsf{periodic\ function}(\log_2 n)$. The average number of leaves is asymptotic to $n/2$; the average (external) path length to $\frac{1}{2}n\log_2 n$; the average degree of the root to 2.

The last paper of Kemp [50] appeared in print shortly after he deceased. It presents a generalized class of balanced trees by introducing *b–balanced trees*: An ordered tree of height h is called b–balanced if all its leaves have a level ℓ with $h-b \leq \ell \leq h$. Kemp computed asymptotic equivalents to the number of all b–balanced trees with n nodes and of all such trees with height h. Furthermore, assuming that all b–balanced ordered trees of size n are equally likely, the average height of such trees together with the corresponding variance is derived.

In [34], a bijection is constructed between the multidimensional binary trees of Section 7 and *monotonically labelled ordered trees.*

These papers of Kemp shed some light on an important component of his personal research programme—Kemp was otherwise not prone to philosophizing on his own results. The general line is that computer science, algorithms and data structures in particular, provides us with a healthy diversity of tree types. Most have quite interesting combinatorial properties. Then, the corresponding enumerative problems should also be of interest *per se*. It is in this light that Kemp's works on digital trees ("tries"), binary search trees, and multidimensional trees are to be understood.

6. Additive weights of trees

Kemp examines additive weights of various kinds of trees [23, 22, 28]. His weights w are recursively defined by

$$w(t) = g(\lambda, n, m) + \sum_{1\le i\le\lambda} \big(a_i(\lambda, n_i, m_i)w(r_i) + f_i(\lambda, n_i, m_i)\big);$$

here, λ is the degree of the root, n the number of nodes, m the number of leaves, i refers to the i–th subtree, and g, a_i, f_i are "simple" auxiliary functions. For instance, upon specialization, one gets parameters like (total, internal, external) path length, total degree path length, number of nodes of a fixed degree and many more. The trees considered are, quite generally, simply generated families of trees, as introduced by Meir and Moon.

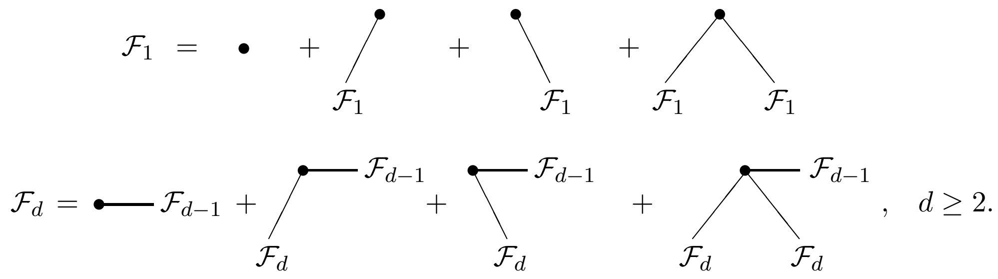
There are many general and special results, too many to be listed here; particular interest is in the average weight over trees from the family with n nodes. In [21] Kemp presents a generalization of his approach where now the trees under consideration can be specified by their total number of nodes n and the number of nodes m_j of degree d_j, $1 \le j \le \ell$. Thus, the former model results from $\ell = 1$ and $d_1 = 0$. In this framework he derives the average behavior of certain path lengths for simply generated families of trees which leads to invariants that are valid for a tree in an arbitrary simply generated family of trees. The paper [25] has a further generalization: instead of a weight function, a system of weight functions is considered.

The research of Kemp in this area shows him largely (but not exclusively) as a formalist. But beyond this aspect, it can be related to more global endeavours, what some of us call "symbolic methods." There the field of investigation is the relation between combinatorial structure and the resulting algebraic structure present in generating functions. Note that the class of models considered are given by the framework of simple varieties of trees (Meir and Moon). Such trees have been gradually recognized to be endowed with an amazingly rich set of combinatorial–analytic properties. That class also coincides with what a branching process produces when conditioning upon the size of the total progeny: this is yet another token of the importance of simple varieties of trees to which Kemp devoted so much of his efforts.

7. Binary search trees

In [27], Kemp considers binary search trees, constructed from d–dimensional keys. The first strategy orders the set of d–dimensional keys lexicographically, and then builds a binary search tree as usual, since 'smaller' resp. 'larger' makes now sense. The second strategy (hierarchical) uses the first entry to insert the tree. If this first entry is already present, then consider the subtree consisting of all keys with that specified first component, and use now the second coordinate to insert the key, etc. The average search times are considered. It is shown that the hierarchical method is never worse than the lexicographical method. Some statistical assumptions have to be made about the input keys; it is not "easy" to find a reasonable model here.

In [36], the *trees* generated by the before mentioned hierarchical algorithm are the major concern, not the keys. They satisfy symbolic equations that are self–explanatory:



The following exact enumeration result is basic: The number $t_d(n_1, \ldots, n_d)$ of such trees with n_s internal nodes in the s–th layer ($1 \leq s \leq d$), is given by

$$t_d(n_1, \ldots, n_d) = \frac{1}{n_d} \prod_{1 \leq p \leq d} \binom{2n_p}{n_p - n_{p-1}}.$$

Note that for $d = 1$, one gets $\frac{1}{n}\binom{2n}{n-1} = \frac{1}{n+1}\binom{2n}{n}$, a familiar Catalan number.

Kemp goes on and considers these *trees* to be equally likely and studies various parameters. The tools here are—perhaps not surprisingly—the Lagrange inversion formula and singularity analysis of generating functions (Kemp invariably prefers the version by Darboux–Pólya). It is interesting to note that the singularities are given by a nonlinear recursion $u_1 = 1/4$ (the classical case), and then $u_p = u_{p-1}/(1 + u_{p-1})^2$.

Kemp's student Uwe Trier wrote his Ph. D. thesis about such topics [S].

In the paper [38], this study is generalized, from binary trees (d–dimensional) to simply generated families of trees (also d–dimensional). Only a master like Kemp could handle such complicated expressions successfully! A further paper [45] deals with d–dimensional binary trees and the enumeration of nodes of 3 types, namely those that have either 0, 1, or 2 subtrees. Kemp obtains very precise and satisfactory results, in particular about the distribution of the nodes within a given layer.

The paper [40] deals with binary search trees, constructed from non–distinct keys. (The usual assumption is that one draws keys from the unit interval.) There are two models: the multiset model, and the probability model. For the typical parameters of binary search trees (path length etc.), Kemp derives precise expressions for the averages, under both models. Note that there is also a revival of interest in these questions: see for instance the highly efficient data structure due to Bentley and Sedgewick [B] and known as the "ternary search tree."

8. Languages, words, and algorithms

For a formal language $\mathcal{L}$, it is a fundamental problem to have algorithms to decide whether a given word w is in $\mathcal{L}$ or not. A general strategy is to scan the word from left to right, until one can stop, because no word in the language has what has been read so far as a prefix, or, alternatively, because one finds that the word *is* in the language.

What is the average length of the shortest prefix which has to be read in order to decide whether or not an input word belongs to the given language? Kemp [41] sets up a general machinery to deal with that question, which of course depends on the given language $\mathcal{L}$. For regular languages, appropriate generating functions can be worked out from the underlying automaton. For the Dyck language (Catalan statistics), this leads to a lengthy series of computations. Further explicit examples are also worked out meticulously, especially some which encode permutations and trees. Some of this is evocative of the "Florentine algorithm," an ingenious technique of Barcucci *et al.* [A] in random generation of combinatorial structures.

The paper [43] is very long and encyclopedic. For various purposes, it is important to have algorithms that generate all combinatorial objects of a given size. This is also related to generating combinatorial objects randomly. Now, there are numerous methods known to code combinatorial objects, especially trees, as certain types of words (Dyck, Motzkin, and many others). If this is the case, then one can rephrase the problem by finding algorithms to go from one word to the next (in the lexicographic sense). Kemp analyzes many of such algorithms, with a precision and detail for which he became extremely famous. Work done recently by Martínez and Molinero [M] in Barcelona has extended somewhat the scope of these ideas, and some of the corresponding algorithms have found their way into combinatorial packages of the symbolic manipulation system Maple.

The subject of random generation and exhaustive listing of combinatorial structures is currently an active one. We start seeing now systematic approaches to the subject. For instance the so–called "recursive method" systematizes early works of Nijenhuis and Wilf, while Boltzmann models, loosely connected to statistical physics, offer promising algorithmic alternatives. It is also significant, at the time when we are writing this note, that Knuth is gradually posting preliminary versions of Volume 4 of *The Art of Computer Programming*. The topics of enumeration and backtracking are central with subjects like "Generating all possibilities" and "Combinatorial generators" being prominent. Surely, our friend Rainer would have enjoyed following these developments and contributing himself to this research stream.

9. The Speckenmeyer papers

Kemp liked to work alone. All his papers are "pure" Kemp, with the exception of two papers published jointly with Speckenmeyer [53, 52]. (One has an additional coauthor, Rosenthal.) Here, Kemp used his expertise to improve greatly on previous material.

Here is the summary of the more general later paper: "The problem of determining whether a Boolean formula in conjunctive normal form is satisfiable in such a way that in each clause exactly one literal is set true and all the other literals are set false is called the exact satisfiability problem. The exact satisfiability problem is well known to be NP–complete and it contains the well–known set partitioning problem as a special case. We study here the average time complexity of a simple backtracking strategy for solving the exact satisfiability problem under two probability models, the constant density model and the constant degree model. For both models we present results sharply separating classes of instances solvable in low degree polynomial time in the average from classes for which superpolynomial or exponential time is needed in the average."

In the introduction to this note, we asserted that Kemp was in several ways a precursor. Here is a proof. His paper with Speckenmeyer [53] was presented at a conference on *Computer Science Logic* (CSL) in 1989. At that time, there had been pretty little work on the precise probabilistic analysis of combinatorial optimization problems. The reference [53] does exactly that for set partitioning. Referring to the abstract that we copied *verbatim* above, it is apparent that Kemp and his coauthors encountered what is now recognized as a *"threshold phenomenon"* or a *"phase transition."* There is immense interest nowadays in these questions as they may help separate "easy" from "hard" instances of difficult combinatorial problems. See for instance recent work by Borgs, Chayes, Dubois, Pittel, and several others—there are even nowadays dedicated conferences and special issues on these topics. This field is even of some industrial relevance given the importance of constraint satisfaction software in a large number of daily life applications.

10. Parsing

The first research of Kemp was related to parsing context–free languages, i.e., to the process of algorithmically determining the derivation tree for a given word and a specific context–free grammar G. One special subclass of context–free grammars are the $LR(k)$ grammars introduced by Knuth. For those grammars it is possible to construct a finite automaton A which acts as finite control of a deterministic pushdown transducer that parses a given input bottom–up. Here, Kemp gave sharp bounds for the size (number of states) of A under various assumptions [1, 2, 4]. For instance he has proven that for G an arbitrary Chomsky reduced grammar, A has at most

$$2 + |\Omega| + \sum_{f \in P} l(Z(f))$$

different states. Here P is the set of productions of G, $Z(f)$ $(l(Z(f)))$ denotes the (length of) the right–hand side of production f and $\Omega := \{w \mid (\exists f, f' \in P)\ (Z(f) = wu,\ Z(f') = wv,\ u \neq v)\}$. Since even the minimal automaton may become rather huge, it is an obvious question to ask whether or not a given parser can be used to parse several languages. This question has been answered by Kemp in [13] where he presented a characterization and a method for the construction of all $LR(0)$ grammars which can be parsed by a given $LR(0)$ parser. It is worth noticing that this number may be infinite. Furthermore, it is decidable whether or not a given $LR(0)$ parser is the (canonical) $LR(0)$ parser of some $LR(0)$ grammar [19]. Later, Kemp considered the same question for a different parsing strategy, the so called precedence matrices. In [26] he characterized all simple precedence grammars which can be parsed by means of a given precedence matrix and presented a method for their construction.

While it is decidable whether or not a given $LR(0)$ parser is the canonical $LR(0)$ parser for some $LR(0)$ grammar, other questions relative to grammars and formal languages are undecidable. For instance we can not decide the ambiguity of a context–free grammar. However, Kemp in [3] gave a criterion which, if satisfied, implies the ambiguity of a context–free grammar. For certain classes of Chomsky reduced context–free grammars this criterion was used to derive estimates for the probability for such a grammar to be ambiguous.

11. The book

Kemp's book *Fundamentals of the average case analysis of particular algorithms* [18] was the first book that *only* dealt with the analysis of algorithms. He discusses models of randomness, permutations and their statistics, random walks and trees, and applications (reduction of binary trees, algorithms to recognise Dyck languages, Batcher's sorting algorithm). There are elaborate appendices on probability theory, grammars and formal power series, generating functions, recurrences, Dirichlet series, Cauchy's integral formula, Euler's summation formula, special combinatorial sequences, and special functions.

The book is deeply rooted in Kemp's own research. There are also many topics of general interest there, and it served its purpose well as the first introduction to the subject (see our comments in our introduction). Today, twenty years later, it is interesting to observe that some of the material it covers is still not to be found in synthetic form anywhere else.

Let us finally mention that in November 2004, there appeared a special issue of the *Journal of the Iranian Mathematical Society* (Volume 3, number 2). The theme is *Probabilistic Analysis of Algorithms*. Thanks to the courtesy of the editors, Professor Ahmad Parsian and Professor Hosam Mahmoud, this special issue has been dedicated to Rainer Kemp "in remembrance of a long and prolific career in analysis of algorithms."

References

[1] R. Kemp. Die Größe des minimalen Analysators einer kontextfreien Grammatik. In *2. Jahrestagung der GI*, volume 78 of *Lecture Notes in Economics and Mathematical Systems*, pages 99–106. Springer, Berlin, 1972.

[2] R. Kemp. An estimation of the set of states of the minimal LR(0)-acceptor. In *Automata, languages and programming (Proc. Sympos., Rocquencourt, 1972)*, pages 563–574. North-Holland, Amsterdam, 1973.

[3] R. Kemp. Mehrdeutigkeiten kontextfreier Grammatiken. In *Automata, languages and programming (Second Colloq., Univ. Saarbrücken, Saarbrücken, 1974)*, pages 534–546. Lecture Notes in Comput. Sci., Vol. 14. Springer, Berlin, 1974.

[4] R. Kemp. Die Größe des zustandsminimalen LR(0)-Analysators. In *Automata theory and formal languages (Second GI Conf., Kaiserslautern, 1975)*, pages 223–232. Lecture Notes in Comput. Sci., Vol. 33. Springer, Berlin, 1975.

[5] R. Kemp. The average number of registers needed to evaluate a binary tree optimally. *Acta Inform.*, 11(4):363–372, 1978/79.

[6] R. Kemp. The average depth of a prefix of the Dycklanguage D_1. In *Fundamentals of computation theory (Proc. Conf. Algebraic, Arith. and Categorical Methods in Comput. Theory, Berlin/Wendisch-Rietz, 1979)*, volume 2 of *Math. Res.*, pages 230–236. Akademie-Verlag, Berlin, 1979.

[7] R. Kemp. The average stack size of a derivation tree generated by a linear context-free grammar. *Inform. and Control*, 42(3):354–365, 1979.

[8] R. Kemp. On the average stack size of regularly distributed binary trees. In *Automata, languages and programming (Sixth Colloq., Graz, 1979)*, volume 71 of *Lecture Notes in Comput. Sci.*, pages 340–355. Springer, Berlin, 1979.

[9] R. Kemp. The average height of r-tuply rooted planted plane trees. *Computing*, 25(3):209–232, 1980.

[10] R. Kemp. A note on the density of inherently ambiguous context-free languages. *Acta Inform.*, 14(3):295–298, 1980.

[11] R. Kemp. A note on the stack size of regularly distributed binary trees. *BIT*, 20(2):157–162, 1980.

[12] R. Kemp. On the average depth of a prefix of a Dycklanguage D_1. *Discrete Math.*, 36(2):155–170, 1981.

[13] R. Kemp. LR(0) grammars generated by LR(0) parsers. *Acta Inform.*, 15(3):265–280, 1981.

[14] R. Kemp. On the average oscillation of a stack. *Combinatorica*, 2(2):157–176, 1982.

[15] R. Kemp. On the number of words in the language $\{w \in \Sigma^* \mid w = w^{\mathrm{R}}\}^2$. *Discrete Math.*, 40(2-3):225–234, 1982.

[16] R. Kemp. The average height of planted plane trees with m leaves. *J. Combin. Theory Ser. B*, 34(2):191–208, 1983.

[17] R. Kemp. The reduction of binary trees by means of an input-restricted deque. *RAIRO Inform. Théor.*, 17(3):249–284, 1983.

[18] R. Kemp. *Fundamentals of the average case analysis of particular algorithms*. Wiley-Teubner Series in Computer Science. John Wiley & Sons Ltd., Chichester, 1984.

[19] R. Kemp. On a decidability and translation result concerning LR(0) grammars. *Elektron. Informationsverarb. Kybernet.*, 20(12):611–621, 1984.

[20] R. Kemp. On a general weight of trees. In *STACS 84 (Paris, 1984)*, volume 166 of *Lecture Notes in Comput. Sci.*, pages 109–120. Springer, Berlin, 1984.

[21] R. Kemp. Free cost measures of trees. In *Fundamentals of computation theory (Cottbus, 1985)*, volume 199 of *Lecture Notes in Comput. Sci.*, pages 175–190. Springer, Berlin, 1985.

[22] R. Kemp. Additive weights of nonregularly distributed trees. In *Random graphs '85 (Poznań, 1985)*, volume 144 of *North-Holland Math. Stud.*, pages 129–155. North-Holland, Amsterdam, 1987.

[23] R. Kemp. The analysis of an additive weight of random trees. *J. Inform. Process. Cybernet.*, 23(10-11):517–528, 1987.

[24] R. Kemp. A note on the number of leftist trees. *Inform. Process. Lett.*, 25(4):227–232, 1987.

[25] R. Kemp. On systems of additive weights of trees. *Fachbereich Informatik, Universität Frankfurt, Interner Bericht 6/87*, 1987.

[26] R. Kemp. A note on precedence analysis. *J. Inform. Process. Cybernet.*, 24(9):401–413, 1988.

[27] R. Kemp. Binary search trees for d-dimensional keys. *J. Inform. Process. Cybernet.*, 25(10):513–527, 1989. Colloquium on Computation Theory (CCT '88) (Berlin, 1988).

[28] R. Kemp. The expected additive weight of trees. *Acta Inform.*, 26(8):711–740, 1989.

[29] R. Kemp. A one-to-one correspondence between two classes of ordered trees. *Inform. Process. Lett.*, 32(5):229–234, 1989.

[30] R. Kemp. Further results on leftist trees. In *Random graphs '87 (Poznań, 1987)*, pages 103–130. Wiley, Chichester, 1990.

[31] R. Kemp. On the number of deepest nodes in ordered trees. *Discrete Math.*, 81(3):247–258, 1990.

[32] R. Kemp. On the stack ramification of binary trees. In *Random graphs, Vol. 2 (Poznań, 1989)*, Wiley-Intersci. Publ., pages 117–138. Wiley, New York, 1992.

[33] R. Kemp. On the stack size of a class of backtrack trees. In *Informatik*, volume 1 of *Teubner-Texte Inform.*, pages 269–282. Teubner, Stuttgart, 1992.

[34] R. Kemp. Monotonically labelled ordered trees and multidimensional binary trees. In *Fundamentals of computation theory (Szeged, 1993)*, volume 710 of *Lecture Notes in Comput. Sci.*, pages 329–341. Springer, Berlin, 1993.

[35] R. Kemp. A note of two classes of t–ary trees. *Elektron. Informationsverarb. Kybernet.*, 29:167–174, 1993.

[36] R. Kemp. Random multidimensional binary trees. *Elektron. Informationsverarb. Kybernet.*, 29:9–36, 1993.

[37] R. Kemp. Balanced ordered trees. In *Proceedings of the Fifth International Seminar on Random Graphs and Probabilistic Methods in Combinatorics and Computer Science (Poznań, 1991)*, volume 5, pages 99–121, 1994.
[38] R. Kemp. On the inner structure of multidimensional simply generated trees. In *Proceedings of the Sixth International Seminar on Random Graphs and Probabilistic Methods in Combinatorics and Computer Science, "Random Graphs '93" (Poznań, 1993)*, volume 6, pages 121–146, 1995.
[39] R. Kemp. Uniformly growing backtrack trees. *RAIRO Inform. Théor. Appl.*, 29(1):45–73, 1995.
[40] R. Kemp. Binary search trees constructed from nondistinct keys with/without specified probabilities. *Theoret. Comput. Sci.*, 156(1-2):39–70, 1996.
[41] R. Kemp. On prefixes of formal languages and their relation to the average-case complexity of the membership problem. *J. Autom. Lang. Comb.*, 1(4):259–303, 1996.
[42] R. Kemp. On the average minimal prefix-length of the generalized semi-Dyck language. *RAIRO Inform. Théor. Appl.*, 30(6):545–561, 1996.
[43] R. Kemp. Generating words lexicographically: an average-case analysis. *Acta Inform.*, 35(1):17–89, 1998.
[44] R. Kemp. On the expected number of nodes at level k in 0-balanced trees. In *STACS 98 (Paris, 1998)*, volume 1373 of *Lecture Notes in Comput. Sci.*, pages 566–576. Springer, Berlin, 1998.
[45] R. Kemp. On the joint distribution of the nodes in uniform multidimensional binary trees. In *Proceedings of the Eighth International Conference "Random Structures and Algorithms" (Poznań, 1997)*, volume 13, pages 261–283, 1998.
[46] R. Kemp. On leftist simply generated trees. *J. Autom. Lang. Comb.*, 4(4):313–331, 1999.
[47] R. Kemp. A one-to-one correspondence between a class of leftist trees and binary trees. *Inform. Process. Lett.*, 71(3-4):97–105, 1999.
[48] R. Kemp. On the expected number of leftist nodes in simply generated trees. *J. Autom. Lang. Comb.*, 6(3):313–343, 2001.
[49] R. Kemp. On the representation of simply generated trees by leftist trees. *European J. Combin.*, 24(5):581–605, 2003.
[50] R. Kemp. On the average height of b-balanced ordered trees. *Journal of the Iranian Statistical Society*, 3(2):175–218, 2004.
[51] R. Kemp and H. Prodinger, editors. *Mathematical analysis of algorithms*. 2001. Algorithmica **31** (2001), no. 3.
[52] J. W. Rosenthal, E. Speckenmeyer, and R. Kemp. Exact satisfiability, a natural extension of set partition, and its average case behavior. *Ann. Math. Artificial Intelligence*, 6(1-3):185–200, 1992.
[53] E. Speckenmeyer and R. Kemp. On the average time complexity of set partitioning. In *CSL '89 (Kaiserslautern, 1989)*, volume 440 of *Lecture Notes in Comput. Sci.*, pages 369–381. Springer, Berlin, 1990.

Additional References

[A] E. Barcucci, R. Pinzani and R. Sprugnoli. The random generation of directed animals. Theoretical Computer Science **127**: 333–350, 1994.
[B] J L. Bentley and R. Sedgewick. Algorithms for Sorting and Searching Strings. Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 1997.
[C] J. Berstel. Sur la densité asymptotique des languages formels. Automata, languages and programming (M. Nivat ed.), North Holland, Amsterdam, 345–358, 1973.
[D] N. G. deBruijn, D. E. Knuth and S. O. Rice. The Average Height of Planted Plane Trees. Graph Theory and Computing (R. C. Read ed.), pp. 15–22, Academic Press, 1972.
[E] A. Ershov. On Programming of Arithmetic Operations. Communications of the ACM **1**: 3–6, 1958.
[F] P. Flajolet, J. C. Raoult and J. Vuillemin. The Number of Registers Required for Evaluating Arithmetic Expressions. Theoretical Computer Science **9**: 99–125, 1979.
[G] M. Hofri. *Analysis of Algorithms*, Oxford University Press, 1987 and 1995.
[H] P. Kirschenhofer and H. Prodinger. On the recursion depth of special tree traversal algorithms. Information and Computation **74**: 15-32, 1987.

[I] D. E. Knuth. The Art of Computer Programming 3: Sorting and Searching, 2nd edition, Addison-Wesley, 1998.

[J] G. Louchard. Gaussian processes and list structures. Theoretical Computer Science **53**: 99–124, 1987.

[K] H. Mahmoud. *Evolution of Random Search Trees*, John Wiley & Sons, New York, 1992.

[L] H. Mahmoud. *Sorting: A Distribution Theory*, John Wiley & Sons, New York, 2000.

[M] C. Martínez and X. Molinero. A generic approach for the unranking of labelled combinatorial classes. Random Structures & Algorithms **19**: 472–497, 2001.

[N] A. Meir and J. Moon. On the altitude of nodes in random trees. Canad. J. Math **30**: 997–1015, 1978.

[O] G. Pólya. Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen, Acta Mathematica **68**: 145–254, 1937.

[P] R. Sedgewick and P. Flajolet. *Analysis of Algorithms*, Addison-Wesley, 1996.

[Q] V. Strehl. Two short proofs of Kemp's identity of rooted plane trees. European J. Combin. **5**: 373–376, 1984.

[R] W. Szpankowski. *Average Case Analysis of Algorithms on Sequences*, John Wiley & Sons, New York, 2001.

[S] U. Trier. Additive weights of a special class of nonuniformly distributed backtrack trees. Inform. Process. Lett. **42**: 67–76, 1992.

[T] I. Yekutieli and B. Mandelbrot. Horton–Strahler ordering of random binary trees. J. Phys. A. **27**: 285–293, 1994.

Philippe Flajolet, Algorithms Project, INRIA-Rocquencourt, F-78153 Le Chesnay, France

E-mail address: `philippe.flajolet@inria.fr`

Markus Nebel, Institut für Informatik, Johann Wolfgang Goethe-Universität Frankfurt, Robert Mayer Str. 11-15, 60325 Frankfurt, Germany

E-mail address: `nebel@sads.informatik.uni-frankfurt.de`

Helmut Prodinger, Mathematics Department, University of Stellenbosch, 7602 Stellenbosch, South Africa

E-mail address: `hproding@sun.ac.za`