[5] Caliebe, A. *Symmetric Fixed Points of a Smoothing Transformation.* Adv. Appl. Probab. **35** (2003), 377–394.

[6] Caliebe, A. *Representation of fixed points of a smoothing transformation.* Mathematics and computer science. III (2004), 311–324, Trends Math., Birkhäuser, Basel.

[7] Caliebe, A. and Rösler, U. *Fixed points with finite variance of a smoothing transformation.* Stochastic Process. Appl. **109** (2003), 105–129.

[8] Rüschendorf, L. *On stochastic recursive equations of sum and max type.* J. Appl. Probab. **43** (2006), 678-703.

[9] Spitzmann, J. *Lösungen inhomogener stochastischer Fixpunktgleichungen (Solutions of inhomogeneous fixed-point equations).* PhD Dissertation, Christian-Albrechts-Universität Kiel.

Towards the Variance of the Profile of Suffix Trees

Mark Daniel Ward
(joint work with Pierre Nicodème)

We consider randomly generated strings from which we (1) determine the profile of the analogous suffix tree, or (2) determine the subword complexity. A suffix tree is a retrieval tree (trie) built from the unique (occurring only once) prefixes of the suffixes of a string. E.g., if $S = 010110011110000100011100 0\ldots$, and if we build a suffix tree from the first 12 strings of S, the 10th suffix has a unique prefix 11000, so it gets inserted as the leaf S_{10} in Figure 2. The suffix tree has "myriad" applications [1].

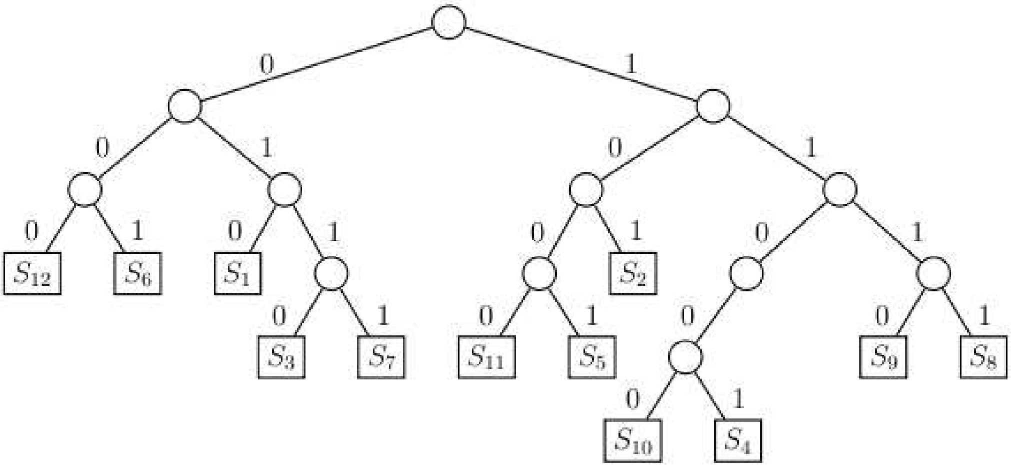

Figure 2. A suffix tree built from string $S = 0101100111100001000111000\ldots$

The (internal) **profile** of a suffix tree at level k is the number of (internal) nodes located on level k. Our goal is to make precise comparisons of the profile of a suffix tree versus the profile of a trie built over independent strings. When the underlying strings all derive from a Bernoulli source, a comparison of the average profile of a suffix tree versus the average profile of a trie built over independent strings was made in [7]. Empirical evidence has been given, however, that the variance of the profile of a suffix tree at level k has asymptotically different behavior than the profile of a trie built over independent strings; see [5]. A recent, comprehensive study of the distribution of the profile of a trie built over independent strings appears in [6].

We use the following **notations**.

- $|S|_w$ is the number of occurrences of the word w in the string S.
- For a set of words $\mathcal{W}_n$ of cardinality n, we write

$$|\mathcal{W}_n|_w = |\{u \in \mathcal{W}_n;\ u = w\}|,\ \text{the number of words of } \mathcal{W}_n \text{ equal to } w.$$

We generate strings randomly over an alphabet $\mathcal{A} = \{a, b\}$ according to a Bernoulli source. In other words, assuming that there are probabilities p and $q = 1 - p$ associated with letters a and b, the probability that a string of length n has exactly j occurrences of a is $\binom{n}{j} p^j q^{n-j}$.

Generating a random string S of length $n + k - 1$ and a set $\mathcal{T}_n$ of n random strings of length k, we consider the boolean indicators

- $I^{(d)}_{n,w} = 1$ if $|S|_w \geq d$ and $I^{(d)}_{n,w} = 0$ elsewhere,
- $J^{(d)}_{n,w} = 1$ if $|\mathcal{T}_n|_w \geq d$ and $J^{(d)}_{n,w} = 0$ elsewhere.

If a suffix tree is built from such a string S, then the profile $X^{(\text{prof})}_{n,k}$ of such a suffix tree is equal to the number of words of length k that occur two or more times as subwords in S. In other words, we observe

$$X^{(\text{prof})}_{n,k} = \sum_{w \in \mathcal{A}^k} I^{(2)}_{n,w},$$

where $\mathcal{A}^k$ is the collection of all binary words of length k.

Similarly, we define

$$Y^{(\text{prof})}_{n,k} = \sum_{w \in \mathcal{A}^k} J^{(2)}_{n,w},$$

which corresponds to the profile of a trie built upon n random strings of length k.

Then [7] proves $X^{(\text{prof})}_{n,k} - Y^{(\text{prof})}_{n,k} = O(n^{-\epsilon}\mu^k)$ for $\epsilon > 0$ and $\mu < 1$, but [5] gives empirical evidence that the variances are asymptotically different.

The ***k*th subword complexity** $X^{(\text{sub})}_{n,k}$ of S (of length $n+k-1$) is the number of distinct subwords of length k that occur at least once as a subword of S. We therefore have

$$X^{(\text{sub})}_{n,k} = \sum_{w \in \mathcal{A}^k} I^{(1)}_{n,w}.$$

Finally, we define

$$Y^{(\text{sub})}_{n,k} = \sum_{w \in \mathcal{A}^k} J^{(1)}_{n,w},$$

where the "sub" is just meant to remind us that $Y^{(\text{sub})}_{n,k}$ is defined similarly to the subword complexity $X^{(\text{sub})}_{n,k}$ above. Then [3] proves $X^{(\text{sub})}_{n,k} - Y^{(\text{sub})}_{n,k} = O(n^{-\epsilon}\mu^k)$

for $\epsilon > 0$ and $\mu < 1$, but empirical evidence (unpublished) also shows that the variances are asymptotically different.

The **correlation set** of a pair of words u, v (here, of the same length) is $\mathcal{C}_{u,v} = \{h \mid u.h = y.v,\ |y| < |u|\}$. The correlation polynomial is the relevant generating function. For example, $u = ababa$ and $v = abaab$ have correlation polynomial $C_{u,v}(z) = P(ab)z^2 + P(baab)z^4$. Previous approaches to problems of this nature use methods by Jacquet, Régnier, Szpankowski, and many others, tracing back to Goulden and Jackson, and Guibas and Odlyzko; here, we use the "cluster" approach that has been initially defined by Goulden and Jackson (see [2] for citations and recent discussion); we also do not consider the relevant complex analysis (this will follow in a longer treatment), but use the following intuitive approach: the primary results will be derived from noting that an autocorrelation polynomial is 1 plus much smaller terms, with high probability, and a correlation polynomial of two distinct words is 0 plus much smaller terms, with high probability; see [4]. Briefly, we have

$$\sum_{n\geq 0} E[Y_{n,k}^{(\mathrm{sub})}]z^n = \sum_{w\in\mathcal{A}^k} (1-(1-P(w)))z^n = \sum_{w\in\mathcal{A}^k} \frac{P(w)z}{(1-z)(1-(1-P(w))z)}.$$

To determine $\sum_{n\geq 0} E[X_{n,k}^{(\mathrm{sub})}]z^n$, we use the cluster approach. The probability generating function for the cluster of a word w is

$$\xi_w(z,t) = \frac{tP(w)z^{|w|}}{1-t(C_w(z)-1)}.$$

The probability generating function for the set of all words, with some of the w's distinguished, is $T_w(z,t) = 1/(1-z-\xi_w(z,t))$. Thus, the probability generating function for the set of words with no occurrences of w is

$$\frac{1}{1-z-\xi(z,-1)} = \frac{C_w(z)}{D_w(z)},$$

where $D_w(z) = (1-z)C_w(z) + P(w)z^{|w|}$. It follows that

$$\sum_{n\geq 0} E[X_{n,k}^{(\mathrm{sub})}]z^n = \sum_{w\in\mathcal{A}^k} \frac{P(w)z}{(1-z)D_w(z)}.$$

These results were first derived in [3], but the cluster approach allows a much more straightforward proof. Clusters allow quick verification of the probability generating functions from [7], and clusters allow the new derivation of the relevant probability generating functions for the variance of the profile of suffix trees and the variance of the subword complexity. MDW has derived several more results in this direction but did not have time to present these derivations during the relatively short talk at MFO. These results will be presented in a longer version of this paper in the near future, and we will complete the analysis using bootstrapping for complex-valued singularities and then using residue analysis.

References

[1] A. Apostolico. The myriad virtues of subword trees. In A. Apostolico and Z. Galil, editors, *Combinatorial Algorithms on Words*, pages 85–95, Berlin, 1985. Springer Verlag.

[2] F. Bassino, J. Clément, and P. Nicodème. Counting occurrences for a finite set of words: combinatorial methods. *ACM Transactions on Algorithms*. To appear.

[3] I. Gheorghiciuc and M. D. Ward. On correlation polynomials and subword complexity. *Discrete Mathematics and Theoretical Computer Science*, AH:1–18, 2007.

[4] P. Jacquet and W. Szpankowski. Autocorrelation on words and its applications. Analysis of suffix trees by string-ruler approach. *Journal of Combinatorial Theory*, A66:237–269, 1994.

[5] P. Nicodème. q-gram analysis and urn models. *Discrete Mathematics and Theoretical Computer Science*, AC:243–258, 2003.

[6] G. Park, H.-K. Hwang, P. Nicodème, and W. Szpankowski. Profiles of tries. *SIAM Journal on Computing*, 38:1821–1880, 2009.

[7] M. D. Ward. The average profile of suffix trees. In *The Fourth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pages 183–193, 2007.

The Quicksort Process

Uwe Rösler

The sorting algorithm Quicksort, invented by Hoare ’61, sorts a given list of n different reals. By now, we have a complete analysis of the running time, including the distribution and large deviation results. Is there an online version of Quicksort in the sense, that given the input of n different numbers, the online version provides first the smallest number, then the second smallest and so on during time. That is very easy to obtain, if we recall Quicksort every time for the list with the smallest numbers. But what about a limit as n tends to infinity as a process?

The answer to this question will be yes, the details will be given in a forthcoming paper by my PhD-student Mohammed Ragab. In this talk we discuss some related problems and technics via the Weighted Branching Process.

Let $X^n(l)$ denote the number of comparisons until the l-th smallest element appears for the online version of Quicksort. Mathematically we can describe the distribution of $X_n(l)$ recursively by

$$X^n(l) \overset{\mathcal{D}}{=} n-1+\mathbb{1}_{I^n\le l}(X_1^{I^n-1}(I^n-1)+X_2^{n-I^n}(l-I^n))+\mathbb{1}_{I^n>l}X_1^{I^n-1}(l)$$

Here I^n, X_i^k for $i=1,2$ and $0\le k<n$ are independent. The distributions of X_1^k, X_2^k, X^k are the same and I^n has the uniform distribution on $\{1,2,\dots,n\}$.

By a result of Martínez, [1], the expectation $a^n(l)=E(X^n(l))$ can be explicitly calculated via the recursion and is

$$a^n(l)=2n+2(n+1)H_n-2(n+3-l)H_{n+1-l}-6l+6$$

where H_n denotes the n-th harmonic number.

The natural normalization

$$Y^n\left(\frac{l}{n}\right)=\frac{X^n(l)-a^n(l)}{n+1}$$