Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Theoretical Computer Science 387 (2007) 197–199
Theoretical Computer Science
www.elsevier.com/locate/tcs

Foreword[☆]

Foreword by the Editors

In 1994, Michael Burrows and David Wheeler published the Technical Report "A block sorting lossless data compression algorithm" describing a new data compression algorithm based on a reversible transformation of the input. This transformation has emerged as a powerful tool in algorithmic design and is today universally known as the BWT: the *Burrows–Wheeler Transform*. The BWT is the heart of bzip2 which is a standard tool for lossless compression. Several years after its discovery, new algorithms and theoretical studies concerning BWT continue to flourish.

The BWT has also revolutionized the field of indexing data structures: using the BWT it is possible to build the so-called "compressed indexes", a new family of data structures which support powerful substring searches using roughly the same space as is used by the best compressors. These results have disproved the belief that an efficient full-text index requires space superlinear in the indexed string length (as for suffix tree and array).

In 2004 we celebrated the 10-year anniversary of the BWT by organizing the DIMACS Workshop "The Burrows–Wheeler Transform: Ten Years Later", with the participation of many researchers and practitioners interested in this fascinating mathematical tool. As a follow up, we edited this volume of research contributions on the BWT and its applications. All contributions were carefully refereed to journal standard.

This volume consists of 11 papers. It starts with three papers (by Fenwick, Kaplan et al., and Giancarlo et al.) regarding properties and analysis of the BWT as a data compression tool. These are followed by two papers (by Karkkainen and Larsson et al.) on algorithmic issues related to the computation of the BWT. Then, we have three papers on innovative applications of the BWT: table compression (Vo et al.), XML compression and indexing (Barbay et al.), and bioinformatics (Mantaci et al.). Finally, the volume includes three papers (Gupta et al., Makinen et al., Golynski) on compressed indexing, which is currently the fastest growing field of application of the BWT.

[☆] The special issue is dedicated to the memory of David Wheeler (1927–2004).

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2007.07.011

Paolo Ferragina
Dipartimento di Informatica,
University of Pisa,
Italy
E-mail address: ferragina@di.unipi.it.

Giovanni Manzini*
Dipartimento di Informatica,
Università del Piemonte Orientale,
Italy
E-mail address: manzini@mfn.unipmn.it.

S. Muthukrishnan
Google Inc.,
New York,
USA
E-mail address: muthu@google.com.

* Corresponding editor. Tel.: +39 0131 360173; fax: +39 0131 360198.

Foreword by Mike Burrows[1]

This foreword gives me the opportunity not only to praise David Wheeler (1927–2004), but also to clarify in print the genesis of the BWT, the compression algorithm that he invented.

David Wheeler was remarkable both for being so smart, and for being so modest. He delighted in inventing new things, but published only the best of them. As a result, few know his name, even though he played a significant role in the development of early computers.

For example, in 1949 as part of Maurice Wilkes' team, he wrote some of the first programs for the EDSAC, the first stored-program computer with more than a few words of memory. The team recognized the importance of subroutines in structuring programs, and to enable this David devised a relocating loader, and an EDSAC instruction sequence to do what we would now recognize as procedure call; this sequence became known as the "Wheeler Jump". He was a major contributor to the first library of subroutines that were invoked in this way, rather than being inserted directly into the code like a macro. Later, David, Maurice Wilkes, and Stanley Gill wrote the first book for programmers; this was a important step in making computers available to non-experts. The programming methodology they advocated looks surprisingly modern even today, even though all of this was done by 1951.

In addition to the EDSAC, David contributed to several other systems that helped shape the future, including ILLIAC, EDSAC 2, Titan, and CAP, and the Cambridge Ring local area network. Another way in which David contributed to the field is in educating others; he taught many undergraduates, and his research students included Roger Needham and Bjarne Stroustrup.

David would sometimes invent a cunning algorithm but would file it away rather than publish it because he felt it was obvious. Many Cambridge graduate students found these ideas less obvious, and had the experience of visiting his office with a problem, only to be given a sheet of paper on which David had written the solution some years previously. This was my experience in 1984, when I became another of David's students. I hate to think of how many of his techniques lay undiscovered in those filing cabinets simply because no student had asked the right question.

David had spent some time visiting Bell Labs, and he invented at least two compression algorithms there. (In the technical report that described the BWT, I gave the year as 1981, but later, with access to the memory of his wife Joyce, we deduced that it must have been 1978.) One of David's algorithms gave modest compression but at high speed, and was based on hashing; fortunately it was independently invented by Raita and Teuhola, so the world got to know of it. The other algorithm was the "block sorting" algorithm, which David considered too slow for routine use, but employed as a benchmark in measuring the effectiveness of other algorithms.

Years passed, and it became clear that David had no thought of publishing the algorithm—he was too busy thinking of new things. Eventually, I decided to force his hand: I could not make him write a paper, but I could write a paper with him, given the right excuse. So I enlisted his help in finding ways to execute the algorithm's sorting step efficiently, which involved considering constant factors as much as asymptotic behavior. We tried many things, only some of which made it into the paper, but we met my goals: we showed that the algorithm could be made fast enough to see practical use on modern machines, and I did enough to assuage my guilt in putting my name next to David's when describing his algorithm.

The Data Compression Conference did not like the paper, perhaps put off by my rather pragmatic presentation. However, Mark Nelson drew attention to it in a Dr. Dobbs article, and that was enough to ensure its survival. I was embarrassed to see that Mark Nelson called it the "Burrows–Wheeler Transform", even though I tried to make it clear that the algorithm was David's. Of course, David was far too polite to mention this.

A wonderful thing about publishing an idea is that a greater number of minds can be brought to bear on the surrounding problems. This issue of Theoretical Computer Science is an example of how an idea can be improved and generalized when more people are involved. I feel sure that David Wheeler would be pleased to see that his technique has inspired so much interesting work.

[1] I am indebted to Maurice Wilkes and Joyce Wheeler for commenting on this foreword. For a first-hand view of some key events in the history of computing, including the EDSAC and some of David Wheeler's early contributions, see Maurice Wilkes' "Memoirs of a Computer Pioneer", MIT Press, 1985, ISBN 0-262-23122-0.